Crowd Knowledge Creation Acceleration

Yang, Jie

DOI
10.4233/uuid:ed22a51a-3469-4699-836d-19322b9537c9

Publication date
2017

Document Version
Final published version

Citation (APA)
Yang, J. (2017). *Crowd Knowledge Creation Acceleration*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:ed22a51a-3469-4699-836d-19322b9537c9

Crowd Knowledge Creation Acceleration

Jie Yang

Crowd Knowledge Creation Acceleration

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op woensdag 15 november 2017 om 10:00 uur

door **Jie YANG**

Master of Science in Computer Science and Engineering,
Technische Universiteit Eindhoven, Nederland
geboren te Wenzhou, Zhejiang, China.

Dit proefschrift is goedgekeurd door de promotoren:
Prof.dr.ir. G.J.P.M. Houben
Copromotor: Dr.ir. A. Bozzon

Samenstelling promotiecommissie:

Rector Magnificus	voorzitter
Prof.dr.ir. G.J.P.M. Houben	Technische Universiteit Delft, promotor
Dr.ir A. Bozzon	Technische Universiteit Delft, co-promotor
Onafhankelijke leden	
Prof.dr. A. Hanjalic	Technische Universiteit Delft
Prof.dr. L.M. Aroyo	VU University Amsterdam
Prof.dr. P. Cudré-Mauroux	University of of Fribourg
Prof.dr. W. Nejdl	Leibniz Universität Hannover
Prof.dr. E. Visser	Technische Universiteit Delft, reservelid

SIKS Dissertation Series No. 2017-47

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Published and distributed by: Jie Yang
E-mail: yangjiera@gmail.com

ISBN: 978-94-6186-865-7

Keywords: Knowledge Creation, Acceleration, Human Computation, Crowdsourcing, Recommender Systems, User Modeling

Cover image: (front) A Barn Owl Flying at British Wildlife Centre, Surrey, England, Creative Commons Attribution; (back) Crowds in Time Square, New York, the United States, by Yuan Lu.
Cover design by: Yuan Lu.

Printed and bound in The Netherlands by Sieca Repro.

Acknowledgments

Upon the completion of this thesis, I would like to deliver my gratitude to all who have given me the guidance, support, advice, help, encouragement, joy, and love that have enabled me to overcome all challenges in pursuing a PhD.

First and foremost, I would like to express my highest gratitude to my promotor Geert-Jan Houben, who led me to the fascinating world of scientific research. Thank you, Geert-Jan, for your thoughtful guidance and strong support during the full period of my PhD. It would not have been possible to write this dissertation without your help. Thank you for the many discussions in our regular meetings as well as those in our daily conversations, from which I have learned so much on both professional and personal development.

I am deeply indebted to my supervisor Alessandro Bozzon, who had also been my officemate. Thank you, Alessandro, for your extensive guidance and sincerest advice that has made this road to PhD a rewarding and enjoyable journey. I cherish the memory of the inspiring and fun discussions we had over a wide range of topics. It was a great pleasure working with you.

I would like to thank the other members in my dissertation committee: Alan Hanjalic, Lora Aroyo, Philippe Cudré-Mauroux, Wolfgang Nejdl, and Eelco Visser, for their insightful feedback.

Many thanks must go to my collaborators: Zhu Sun, Jie Zhang, Claudia Hauff, Judith Redi, Gianluca Demartini, Ujwal Gadiraju, Thomas Drake, Andreas Damianou, Yoelle Maarek, Wenjie Pei, Lora Aroyo, Martha Larson, Cynthia Liem, Andrea Tagarelli, Guanliang Chen, Ke Tao, Jasper Oosterman, Achilleas Psyllidis, Vincent Gong, Christiaan Titos Bolivar, Deniz Iren, and Tobias Hoßfeld. It was truly honorable and enjoyable to work with all of them. I also thank the former master students for their participation in my PhD project: Carlo van der Valk, Giuseppe Silvestri, Arkka Dhiratara, Friso Abcouwer, and Sijmen Hoogendijk. Special thanks go to my former graduate advisors, Toon Calders, Paul De Bra, Xi Long, Hoang Thanh Lam, and Reinder Haakma, for their advice on my master thesis and recommendation for my PhD application.

I am grateful to the members of the Web Information Systems (WIS) group and the former WISers for their help and friendship: Claudia Hauff, Jan Hidders, Christoph Lofi, Nava Tintarev, Asterios Katsifodimos, Stefano Bocconi, Achilleas Psyllidis, Pavel Kucherbaev, Mohammad Khalil, Qi Gao, Ke Tao, Jasper Oosterman, Guanliang Chen, Yue Zhao, Dan Davis, Sepideh Mesbah, Vincent Gong, Felipe Moraes, Shahin Sharifi, and Sihang Qiu.

I would like to thank Thomas Drake, Emilio Maldonado, and Yoelle Maarek for the productive and inspiring internship in the Alexa Shopping Machine Learning team in Amazon and for the continued collaboration. Thanks also go to my colleagues: Rongting Zhang, Yanbin Lv, Jiexian Li, Swati Adhikala, Julia Reinspach, Karen Hovsepian, Simone Filice, Theodoros Vasiloudis, and Trang Tran.

I owe thanks to my Chinese friends: Hui Jiang, Yan Jin, Xiangrong Wang, Xuefei Chen, for bringing enormous joy to my life. A special thanks go to Wenjie Pei who has been a great friend and collaborator since my master program.

I would like to express my sincere gratitude to Yuan Lu. Thank you, Yuanyuan, I have been so lucky to have your company and support during the last four years.

Last but not least, I would like to express my deepest gratitude to my mother, Cuilan Lin, for her unconditional love. My utmost gratitude goes to my father, Qingguang Yang, for his support and the non-stopping encouragement, even in the last few months of his life.

Jie Yang
October 2017
Rotterdam, the Netherlands

Contents

Chapter 1

Introduction

1.1 Motivation

Information systems are highly useful in many scenarios where *knowledge* about the world is to be stored and automatically processed for problem-solving. They have extensive applications, ranging from specific domains, e.g., theMednet.org[1] for the clinical domain and StackOverflow[2] for software engineering, to the open domain, e.g., Knowledge Graph[3] by Google for general information retrieval. However, knowledge is a scarce resource, as it can only be acquired through education or experience. Knowledge creation, defined as the process of generating and encoding knowledge into knowledge repositories, is therefore a key step to develop information systems for many applications [28, 174].

The main bottleneck of knowledge creation has been *scalability*. Knowledge creation used to be performed by a small group of domain experts. The process therefore involved domain expert recruitment, and afterwards, knowledge acquisition through interviews or questionnaires. Such a process is costly and time-consuming. As a consequence, knowledge repositories created in this way are limited in terms of the amount of generated knowledge.

Computer science researchers have therefore been pursuing techniques to enable knowledge creation *at scale*. On the one hand, research has been devoted to develop more intelligent *machines* to automatically distill knowledge from data. For example, deep neural network approaches have been devel-

[1] https://www.themednet.org
[2] http://stackoverflow.com
[3] https://www.google.com/intl/es419/insidesearch/features/search/knowledge.html

oped for extracting from unstructured text named entities and the stated relationships between such entities [244]. These techniques, while having made considerable progress, suffer from two major limitations. First, instead of generating *new* knowledge, they can only extract knowledge from existing data. Second, these methods are still *far from accurate* in many knowledge creation tasks where large amount of data are not available. For example, when extracting entities of novel types for which only small datasets are available (e.g., 1000 instances), state-of-the-art neural network based methods can only reach an F1-measure of 0.6 [175].

On the other hand, researchers have been investigating techniques to enable large scale knowledge creation by exploiting *crowds*. In fact, humans are more capable than machines in knowledge creation. Tasks suitable for machines to execute are mainly those that are abstract and formal, e.g., playing chess [101]. Such tasks, while requiring large amount of computation, are not difficult in the sense that they can be completely described by a brief list of formal rules [78]. In contrast, machines are not good at many tasks humans excel at, e.g., conversation, object recognition from pictures or videos, etc. These tasks usually require subjective, intuitive, or specialist knowledge, which is possessed by most or some individuals.

Existing *crowd knowledge creation systems* can generally be categorized into two types: *1)* on-line knowledge crowdsourcing systems, including Wikipedia[4] and community question-answering (CQA) systems such as StackOverflow and Quora[5]; and *2)* human computation systems, such as Amazon Mechanical Turk[6] (mTurk) and CrowdFlower[7]. In these two types of systems, knowledge can be created at scale by groups of individuals (e.g., contributors in on-line knowledge crowdsourcing systems, workers in human computation systems) executing corresponding tasks [164] (e.g., questions in CQA systems, micro-tasks in human computation systems). Thanks to the development of Web technologies and the recent development of human computation and crowdsourcing techniques [100, 127, 29, 31], many of these systems have achieved quite a success.

Despite that, each of the two types of systems suffer from their own *limitations*. On-line knowledge crowdsourcing systems are usually oriented at more complex tasks for specialist knowledge generation. However, tasks in these systems are generally solved as a bottom-up process that is largely

[4]https://www.wikipedia.org
[5]https://www.quora.com
[6]https://www.mturk.com/mturk/welcome
[7]https://www.crowdflower.com

uncontrolled. As a result, the outcomes are heavily dependent on the spontaneous and autonomous contribution of crowds. This limits our ability to control the amount, speed, and quality of the generated knowledge. Human computation systems, on the other hand, are usually more controlled with certain guarantees on the amount of task executions and the execution time. However, tasks in these systems are of low complexity. Knowledge creation in these systems only exploits *availability* as the relevant worker property. These systems therefore cannot fully capitalize on other properties that are important for high-quality knowledge creation, e.g., expertise.

By filling the *gap* between the two types of crowd knowledge creation systems, we envision that crowd knowledge creation in the future can unlock the full potential of human cognitive capabilities to solve complex, cognitive intensive tasks, to efficiently generate high-quality knowledge.

1.2 Objectives

This thesis aims at understanding crowd knowledge creation processes to develop methods and tools for controlling and accelerating the process. In short, we formulate this goal as *crowd knowledge creation acceleration*, where we use "acceleration" to describe both the improvement of the speed of knowledge creation and the quality of the generated knowledge.

To capture the key steps in crowd knowledge creation, so as to decompose our objective, we frame the discussion around a generic model that describes the process. As illustrated in Figure 1.1, the model builds on the following key components: ***1)*** *Crowd modeling*, to assess crowd knowledge-related features; ***2)*** *Task modeling*, to represent knowledge demands and resources for knowledge creation; ***3)*** *Task assignment*, for associating tasks with crowds. These components correspond to the key facets of crowd computing systems: component 1 considers crowd properties and engagement; component 2 defines the goal of a task and resources for knowledge creation; with proper crowd and task modeling, component 3 then associates tasks to the right crowds for accelerated knowledge creation.

Correspondingly to the above model, our work aims at developing methods and tools for crowd modeling, task modeling, and task assignment. Our work aims at showing how, by optimally designing each of these techniques, it will be possible to accelerate crowd knowledge creation in a principled and effective way.

In summary, this thesis makes the following research contributions.

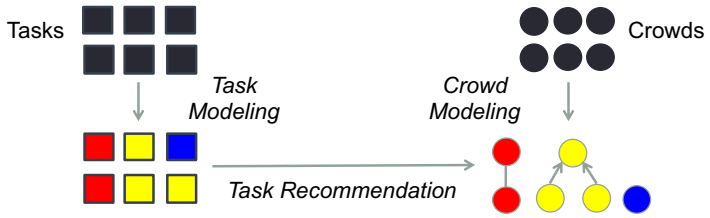


Figure 1.1: Unified model for crowd knowledge creation acceleration.

- **Crowd Modeling Techniques.** We contribute a methodology for principled characterization of expertise based on individual performance in knowledge creation, social interactions, and other related activities. We provide insights on how crowd activities influence and are influenced by knowledge creation marketplaces.
- **Task Modeling Techniques.** We contribute novel methods for quantifying the quality of task formulation, and measuring task complexity and clarity, based on task properties. We contribute insights of how these properties affect task completion rates. We further provide guidelines for better task design, so as to enhance knowledge creation.
- **Task Assignment Methods.** We contribute novel task assignment methods that account for both properties of crowds and tasks. By formulating task assignment as a recommendation problem, we further push forward the field of recommendation by contributing state-of-the-art methods that fully exploit the structure of properties of crowds and tasks.

1.3 Thesis Outline

The thesis contains twelve chapters. After introducing the motivation and the objective in the present chapter, the main body of the thesis contains three parts, each addressing an individual component of our unified model. Each part contains three to four chapters, focusing on different, yet connected aspects of the corresponding component. Each chapter will start with the main research challenge, positioned in existing literature. The challenge is then addressed either by empirical studies or algorithmic design, concluded with findings and contributions.

Part I introduces our work on crowd modeling. We study individual and social aspects of expertise, its multifaceted traits manifested in various ac-

tivities across multiple social networks, and the relationships between crowd community activities and the dynamics of knowledge creation marketplaces.

In **Chapter 2**, we study expertise characterization in community question-answering (CQA) systems. Inspired by the theories of expertise in sociology, we propose a novel expertise metric based on social judgment, namely the Mean Expertise Contribution (MEC). Through empirical study we show that MEC can better characterize expertise than traditional metrics that are biased towards activeness. We then conduct a large-scale analysis to find out how experts identified by MEC behave differently from other crowds in knowledge creation activities.

In **Chapter 3**, we extend our study to multifaceted traits of expertise, namely, specialist expertise and ubiquitous expertise. We investigate how these expertise traits manifest differently across multiple social networks in knowledge creation, sharing of resources, and social interactions. We further analyze how expertise plays a role in the formation and activities of crowd communities. To demonstrate the benefit of cross-platform expertise characterization, we address the problem of improving knowledge creation processes in CQA systems.

In **Chapter 4**, we switch our focus from individuals to crowd workers as communities. We investigate their social behaviors and the relationship with the dynamics of crowdsourcing matketplaces. We performed an analysis on the linkage between crowds' discussions in fora and task executions in crowdsourcing marketplaces, to uncover distinctive crowd preferences in knowledge creation. We then quantify the effect of crowd discussions on task completion rates in marketplaces, thus to show how activities of crowd communities can have a significant effect on task performance.

Part II focuses on task modeling. We study a set of task properties that can be related to the quality and speed of their executions by crowds, namely: the quality of task formulation, task complexity, and clarity of task presentation.

In **Chapter 5**, we first analyze the quality of task formulation in CQA systems. We observe a large portion of poorly formulated tasks. Through a qualitative study, we categorize task formulations of poor quality. To assist askers in task formulation, we then propose methods for automatically detecting whether or not a task is poorly formulated, and if so, suggesting which type of editing actions are required to improve task formulation quality.

In **Chapter 6**, we study task complexity. Given that complexity is a subjective property, we first conduct experiments to understand whether or not complexity is perceived coherently by crowds. We then analyze how com-

plexity can be affected by task types and task design features, such as metadata features (e.g., reward), content description, and the visual design (e.g., colourfulness). We propose a method based on these features for complexity measurement. Finally, we demonstrate the utility of complexity features in predicting task performance.

In **Chapter 7**, we investigate the role of task clarity in crowdsourcing. We first verify the presence of issues with task clarity by surveying workers. Next, we analyze the relationships between two clarity constructs, namely the goal and role clarity. Based on a set of tasks spanning over one year's Amazon mTurk data, we collect crowds' assessment on task clarity, to understand how clarity is perceived by workers. We then propose a set of task features and an automatic method built on these features to measure task clarity.

Part III addresses the problem of assigning tasks to crowds. We formalize task assignment as a recommendation problem, and design novel recommendation methods to fully exploit crowd and task properties for optimal task assignment.

In **Chapter 8**, we investigate the effect of knowledge-related features of the crowd and task topics on task recommendation. Specifically, we analyze the effect of interests and expertise of crowds, together with their intrinsic and extrinsic motivation on task recommendation. Moreover, we analyze the correlation among these features across task topics. We then propose a learning-to-rank based method that accounts for the aforementioned crowd features and task topics for task recommendation.

In **Chapter 9**, we study the structured nature of crowd and task properties, which are often organized in taxonomies. We analyze multiple recommendation datasets, to explore how crowd and task similarity can be induced from their structured properties. We then design a novel regularization technique to model such similarity, namely recursive regularization. Next we propose a novel recommendation method, i.e., ReMF, which integrates recursive regularization into the widely used latent factor model to improve recommendation performance.

In **Chapter 10**, we identify other two types of relationships of crowds and tasks that could be induced from their structure properties, namely, complementarity and alternativity. We propose metrics to capture these relationships, and conduct empirical analysis to verify the presence of these relationships in multiple datasets. We then design a novel recommendation method, namely HieVH, that seamlessly fuses these two relationships into

the latent factor model for improving recommendation performance and interpretability.

In **Chapter 11**, we look into neural network based methods with an aim to learn better representations of crowds and tasks for task assignment. We first adapt the general representation learning method to enable personalized ranking for recommendation. Following the previous chapters, we then design a unified Bayesian framework, i.e., MRLR, that integrates personalized ranking with structured properties of crowds and tasks for representation learning. We analyze the representations learned by MRLR to provide insights on how it can improve recommendation performance.

1.4 Origin of Chapters

The present chapter is based on a doctoral symposium paper. All the main chapters (Chapter 2-11) in this thesis are based on existing papers. Except chapter 3 and chapter 4 which are new contents, all the other chapters have been published as full research papers in conferences related to the research topics of this thesis.

- **Chapter 1** is based on the doctoral symposium paper published at the 15th International Conference on Web Engineering (ICWE 2015) [231].
- **Part I: Crowd Modeling.**
  - **Chapter 2** is based on the paper published at the 22nd International Conference on User Modeling, Adaptation, and Personalization (UMAP 2014) [229].
  - **Chapter 3** contains new research work.
  - **Chapter 4** contains new research work.
- **Part II: Task Modeling.**
  - **Chapter 5** is based on the paper published at the 25th ACM conference on Hypertext and Social Media (ACM HT 2014) [228].
  - **Chapter 6** is based on the paper published at the 4th AAAI Conference on Human Computation and Crowdsourcing (AAAI HCOMP 2016) [232].
  - **Chapter 7** is based on the paper published at the 28th ACM Conference on Hypertext and Social Media (ACM HT 2017) [72].

- **Part III: Task Assignment.**
  - **Chapter 8** is based on the paper published at the 23rd International Conference on User Modeling, Adaptation, and Personalization (UMAP 2015) [230].
  - **Chapter 9** is based on the paper published at the 10th ACM Conference on Recommender Systems (ACM RecSys 2016) [233].
  - **Chapter 10** is based on the paper published at the 31st AAAI Conference on Artificial Intelligence (AAAI 2017) [207].
  - **Chapter 11** is based on the paper published at the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017) [208].

Finally, Chapter 12 concludes this thesis by summarizing the main findings and contributions. Based on them, we provide an outlook to future research directions in related fields.

Part I

Crowd Modeling

This part introduces our work on crowd modeling. We begin our investigations on the concept of expertise, a highly important property of crowds for the purpose of generating high-quality knowledge. The demand for expertise is the basic characteristic that distinguishes tasks in on-line knowledge crowdsourcing systems from those in human computation systems. To fill this gap between these two types of crowd knowledge creation systems, and to reach our ultimate goal of crowd knowledge creation acceleration, this chapter takes the first step to characterize expertise in on-line knowledge crowdsourcing systems, so as to understand the current status, limitation, and potential of expertise usage in crowd knowledge creation.

Chapter 2. Our study starts with the following observation: knownledge creation activities in on-line knowledge crowdsourcing systems are dominated by a relatively small subset of active crowds, due to the built-in incentivization mechanisms. Such a phenomenon poses a big challenge for expertise identification. That is, when improperly designed, it can easily misjudge activeness for expertise. Inspired by the theories of expertise in sociology, we propose a novel expertise metric based on social judgment, namely the Mean Expertise Contribution (MEC). Through empirical study, we show that MEC can better characterize expertise than traditional metrics. Our study further reveals the highly different behaviors between experts and active crowds, showing that experts contribute much less than active crowds, and are much less affected by incentivization mechanisms.

Chapter 3. To further our understanding of expertise, we extend the study to the more refined traits of expertise, namely specialist expertise and ubiquitous expertise. Driven by the fact that expertise is usually created in an interactive process, we further investigate the manifestation of expertise in a multitude of (social) activities of crowds, including production and sharing of resources, and social interactions with each other. We thus provide a principled characterization of expertise along both dimensions, i.e., expertise traits and their manifestations in individual and social activities. To demonstrate the benefit of a principled characterization of expertise, we address the problem of question routing in community question-answering systems. We show that different expertise traits can help finding different types of contributors that best fit different knowledge creation tasks.

Chapter 4. To exploit crowd expertise for knowledge creation acceleration, it further requires to understand the relationships between crowd preferences and knowledge creation demand and outcomes. Given the coupled datasets of crowd discussions in fora and task availability and executions in marketplaces, human computation systems provide great opportunities to study

such relationships. Through the analysis on a 6-years worth of data, we show certain differences in crowd preferences and knowledge demand. However more importantly, we find strong evidences that show the mutual influence between crowd discussions and market dynamics. Specifically, we find that the increasing availability of tasks in the marketplace can trigger crowd discussions. On the other hand, discussions on tasks by crowd communities can positively affect task performance.

Overall, this part contributes new approaches for expertise characterization and understanding on the relationships between crowd preferences and task executions, to fully exploit crowds for knowledge creation acceleration.

Chapter 2

Sparrows and Owls: Expertise Characterization

In this chapter, we study expertise characterization in on-line knowledge crowdsourcing systems. Specifically, we analyze community question-answering (CQA) systems, in which we use "users" to refer to the more general concept of crowds. We introduce a novel expertise metric, i.e., Mean Expertise Contribution (MEC), and conduct a large-scale data analysis to verify the effectiveness in capturing expertise. We further show the distinct behavior of experts (referred to as owls) in contrast to highly active crowds (referred to as sparrows).

This chapter is published as "Sparrows and Owls: Characterization of Expert Behaviour in StackOverflow" [229], by J. Yang, K. Tao, A. Bozzon, and G.-J. Houben in Proceedings of the User Modeling, Adaption and Personalization Conference, pages 266-277. Springer, 2014.

2.1 Introduction

Community question-answering (CQA) platforms like Yahoo! Answers or StackExchange are an important class of social Web applications. Users access such platforms: 1) to look for existing solutions to their issues; 2) to post a new question to the platform community; 3) to contribute by providing new answers; or 4) to comment or vote existing questions and answers. As a result, users jointly contribute to the creation of evolving, crowdsourced, and peer-assessed knowledge bases.

To foster participation, CQA platforms employ effective gamification mechanisms [10] that motivate users by showing a public *reputation score* (calculated by summing the number of preferences obtained by all the posted questions and answers), and by assigning *badges* after achieving pre-defined goals (e.g. complete at least one review task, achieve a score of 100 or more for an answer).

As shown in several studies, CQA platforms are fuelled by a set of highly active users that, alone, contributes to the vast majority of the produced content. Such users, that we call *sparrows*, are clearly an important component of a CQA ecosystem: as their name suggests, they are numerous, highly active, and highly "social" users. However, *sparrows* are not necessarily functional to knowledge creation. Being driven by the gamification incentives, their goal might not be to provide a thorough answer to a question, but simply to "add up" reputation score. To this end, their answers, while quantitatively relevant, might be of low quality and/or low utility (i.e. having low scores from other users and/or ranked low among all the answers in a question); also, to minimize their effort, they might target simple or non-relevant questions.

Sparrows can guarantee responsive and constant feedback, thus playing an important role in keeping the community alive. However, we claim that there exists another category of users having comparable, if not greater importance. Such a category, that we call *owls*, contains users that, while being active members of the community, are driven by another motivation: to increase the overall knowledge contained in the platform. *Owls* are **experts** in the discussed topic, and they prove their expertise by providing useful answers, possibly to questions that are perceived as important or difficult by the community.

Previous studies focused on the characterization of experts in CQA platforms [89, 168, 169]. However, existing methods for expert identification mainly targeted *sparrows*, as they focused on quantitative properties of users'

activities (e.g. reputation score, number of answers) while ignoring the inflationary effect that gamification incentives could trigger.

This chapter targets StackOverflow, a question answering system specialized in software-related issues, and provides two main contributions: 1) a novel expertise assessment metric, called `MEC` (Mean Expertise Contribution), which helps in better discriminating *owls* from *sparrows* and normal users in CQA platforms; and 2) a comparative study of the behaviour of *owls* and *sparrows* in StackOverflow. With respect to the second contribution, we address the following research questions:

- **RQ1**: How do *owls* and *sparrows* differ in terms of knowledge creation and community participation behaviours?
- **RQ2**: How do the overall activities of *owls* and *sparrows* evolve over time?

Understanding the nature of experts, their activity behaviour, and their role is of fundamental importance to drive the economy and prosperity of this class of systems. A better characterization of the quality of users' contributions can also help in improving the performance of user modeling, expert retrieval, and question recommendation systems. Moreover, CQA platforms can develop targeted motivation, engagement, and retention policies specifically addressed to different type of contributors, thus maximising their effectiveness. Finally, companies can better elicit the actual expertise of a potential employee, by exploiting a more accurate characterization of their social reputation. Although the study specifically focused on StackOverflow, we believe that our results are of general interest to crowd knowledge creation.

The remainder of the chapter is organized as follows: Section 2.2 briefly introduces the dataset used in our study. Section 2.3 describes and evaluates the new `MEC` metric. Section 2.4 compares the behaviour of *owls* and *sparrows*. Section 2.5 describes related work, before Section 2.6 presents our conclusions.

2.2 Dataset Description

Launched in 2008, StackOverflow is one of the dominant domain-specific CQA systems on the Web: with 2.3M users, 5.6M active questions, 10.3M an-

swers, and 22.7M comments, StackOverflow[2] aims at becoming a very broad knowledge base for software developers, and it adopts a peer-reviewed moderation policy to close or remove duplicate and off-topic questions. Questions are topically classified by their submitter using one or more *tags*.

Definitions Given a topic t, we define: 1) Q_t as the set of all t-related questions. 2) A_t as the set of all t-related answers; 3) U_t as all the users that participate in discussions about t; 4) A_t^u as the set of answers provided by a user $u \in U_t$ for topic t; 5) Q_t^u as the set of questions answered by user $u \in U_t$ for topic t; 6) $A_{q,t}$ as the set of answers provided for the question $q \in Q_t$ for topic t.

A question $q \in Q_t$ is associated with an owner $u_q \in U_t$, the content c_q, the timestamp of creation ts_q, and the number of views v_q. Similarly, an answer $a \in A_t$ is described by its creator $u_a \in U_t$, content c_a, the timestamp of creation ts_a, and the number of votes it received v_a.

Description	Characteristic
Number of questions	472,860
Number of answers	1,071,750
Number of answerers	117,113
Average voting scores $a_t \in A_t$	2.18±7.35
Average number of answers to question $q_t \in Q_t$	2.27±1.74
Average number of answers given by user $u_t \in U_t$	9.15±76.66

Table 2.1: Descriptive statistics about users activity for the C# topic.

Table 2.1 reports some descriptive statistics related to the topic C#, the most discussed topic in StackOverflow. It clearly emerges a strongly biased distribution in the number of answers provided by each user. Fig. 2.1 plots on a `log-log` scale the distribution of number of answers per question, and number of answers per users in the C# topics. Both quantities resemble a power-law distribution. Fig. 2.2 clearly shows that there are a few users giving many answers.

This is a property that is exhibited by the whole StackOverflow platform, where the most 13% active users, which provided at least ≥ 10 answers, are responsible for 87% of all the answers. We refer to such users as ***Sparrows***, i.e. users that, for a given topic, have $|A_{u,t}| \geq 10$.

[2]The dataset can be accessed at https://archive.org/details/stackexchange. Our study is based on data created up until September 2013.

2.3 Expertise Metric

An expert can be defined as someone who is recognized to be skilful and/or knowledgable in some specific field [66], according to the judgment of the public or his or her peers; expertise then refers to the characteristics, skills, and knowledge that distinguish experts from novices and less experienced people.

In the context of a CQA system, social judgement is critical for expert identification. A question is usually answered by a set of users, whose answers are voted up or down by other members of the platform. On the one hand, answering questions reflects a user's capability of applying knowledge to solve problems. On the other hand, the *voting* from other users can be viewed as a cyber simulation of *social judgement* for the answerers' expertise level.

Note that asking a question and posting a comment may also provide evidence of a user's expertise. However since answering a question can *directly* reflect the knowledge of a user in solving real problems – i.e., actionable knowledge – we limit our discussion of expertise judgement within the scope of answerers. Such choice is also aligned with previous studies of expert identification on CQA systems [26, 168, 169, 240].

2.3.1 Characterization of Expertise

Previous works related expertise to the overall activeness of users in the platform. A classical and often used metric of expertise is the $Z_{Score} = \frac{a-q}{\sqrt{a+q}}$ [240], which measures users according to the number of posted questions q and answers a. Alternatively, one can look at the *reputation* of the user as

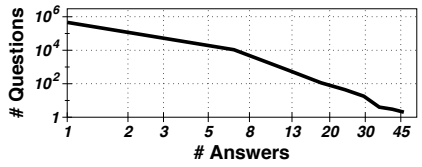


Figure 2.1: C# topic: distribution of number of answers per question.

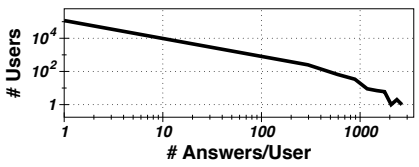


Figure 2.2: C# topic: distribution of number of answers per user.

calculated by the platform [89, 168], a metric that is highly correlated with the number of provided answers.[3]

These two measures suffer from a common problem: they are heavily biased towards user activeness, thus favouring highly engaged users – the *sparrows* – over the ones that provide high level contributions – the *owls*. To support our claim, we performed an analysis of the distribution of the quality of users contribution for *C#*. We considered two dimensions:

1. The **debatableness** of a question, measured according to the *number of answers* it generated;
2. The **utility** of an answer, measured according to its relative *rank* in the list of answers.

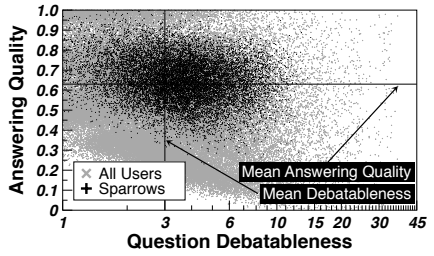


Figure 2.3: Distribution of users according to the avg. debatableness of questions they answer, and the avg. answer quality. *Sparrows*: users with $|A_{u,t}| \geq 10$.

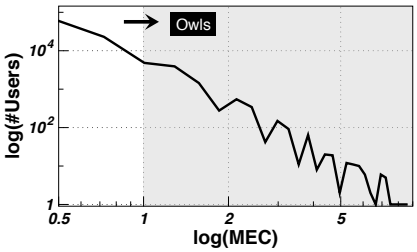


Figure 2.4: Distribution of `MEC` (Mean Expertise Contribution) values in the considered user population. *Owls*: users with `MEC` ≥ 1.

Intuitively, difficult questions generate a lot of discussions, and several answers; also, the higher in the rank an answer has been voted, the more potentially useful it is to solve the related question, and the more it provides evidences about the expertise of the answerer in the topic. Table 2.2 contains a representative example[4] of debatable StackOverflow question. 13 out of 14 answers were provided by very active users, but the best answer was given by a user with only 2 questions answered.

Such phenomenon is not rare, as shown in Fig. 2.3, which visualizes the entire C# dataset. Each dot represents one of the $\sim$ 117K users that provided

[3]For instance, the Spearman correlation between user reputation and total number of answers given by users in topic C# is 0.68.

[4]This question can be accessed at http://stackoverflow.com/questions/21475723

Question: *C# to C++ 'Gotchas'.*		
Rank	**Content**	**#Answered questions***
1st	*C++ has so many gotchas...*	2 answered questions
2nd	*Garbage collection!*	26 answered questions
3rd	*There are a lot of differences...*	175 answered questions
...	...	...
14th	*The following isn't meant*	24 answered questions

*This column shows the number of historical answers to C# questions by the corresponding answerer.

Table 2.2: An example question to which all answers were provided by sparrows except the best answer.

at least one answer for the C# topics. A user is described by the average **utility** of his/her answers (a value in the $[0, 1]$, where 1 represents maximum utility), and by the average **debatableness** of the questions he/she contributed to. The $\sim$ 15K *Sparrows* are highlighted with black crosses. An evident phenomenon can be observed: the vast majority of users answers less debated questions, while only a few (approximately 10%) are able to consistently provide relevant contributions to highly debated questions. Only a fraction ($\sim$30%) of the *sparrows* belongs to the latter group, clearly showing how activeness does not suffice as a measure of expertise.

2.3.2 Identifying Experts

To better identify expert users, we devise a novel strategy for expertise judgement called `MEC` (Mean Expertise Contribution). Differently from existing measures, `MEC` values three expertise factors, namely: answering quality, question debatableness, and user activeness. `MEC` relates to a given topic t, and it is defined as:

$$\texttt{MEC}_{u,t} = \frac{1}{|Q_t^u|} \sum_{\forall q_i \in Q_{u,t}} \mathcal{AU}(u, q_i) * \frac{\mathcal{D}(q_i)}{\mathcal{D}_t^{avg}}$$

where:

- $\mathcal{AU}(u, q_i)$ is the **utility** of the answer provided by user u to question q_i; in our study, $\mathcal{AU}(u, q_i) = \frac{1}{Rank(a_{q_i})}$, that is the inverse of the rank of the answer provided by u for question q. The larger $\mathcal{AU}$, the higher the expertise level shown by the user in question q_i;
- $\mathcal{D}$ is the **debatableness** of the question q_i, calculated as the number of answers $|A_{q_i,t}|$ provided for question q_i;

- $\mathcal{D}_t^{avg}$ is the **average debatableness** of all the questions related to the topic t, calculated as $\frac{1}{|Q_t|} * \sum_{\forall q_j \in Q_t} |A_{q_j,t}|$.

The use of the inverse rank of a question allows to capture the quality of an answer regardless of the judgment expressed by the question provider: indeed, a requester can accept an answer as the right one, although the community, in the long run, might have a different opinion. The sum-up value of the **utility** of the provided answers acts as an indication of the expertise level of a user in a topic. By weighting in the relative debatableness questions, MEC accounts for the average difficulty of questions about a given topic. Note that $\mathcal{AU}(u, q_i) * \mathcal{D}(q_i)$ can be interpreted as the inversed *relative ranking* of u's answer among all answers to question q_i. To factor out user activeness, the resulting value is normalized over the total number of answers a user gave.

A value of $\texttt{MEC}_{u,t} = 1$ indicates that the user u, on average, provides the best answer to averagely debated questions, while $\texttt{MEC}_{u,t} = 0.5$ indicates that u ranks second in answering averagely debated questions, or ranks first in answering less debatable questions.

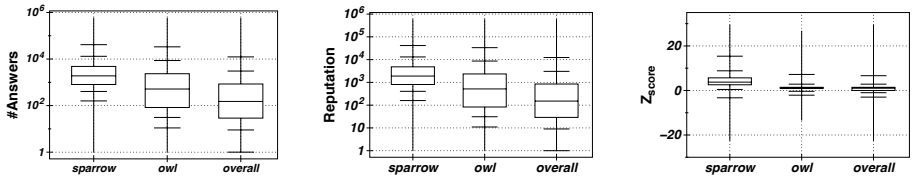
Fig. 2.4 depicts the log-log scale distribution of MEC w.r.t. the population of users involved in the C# topic. Only 11,910 users (approximately 10%) possess a $\texttt{MEC} \geq 1$: we refer to such users as ***Owls***, and observe that for the considered topic their number is significantly lower than the number of *sparrows*.

Fig. 2.5 shows the characterization in terms of number of answers, reputation, and Z_{Score} of *sparrows*, *owls*, and the overall population: *sparrows* consistently obtain higher values, thus erroneously taken as experts. By conservatively considering only the *sparrows* classifying in the top 10% according to number of answers, reputation, and Z_{Score}, we observe that, respectively, only the 9.9%, 21.9% and 10.2% of them also belong to the set of *owls* (i.e. $\texttt{MEC} \geq 1$).

In the following sections we will delve into more details about the different nature of *owls* and *sparrows*, highlighting their divergent behaviours and roles in StackOverflow.

2.4 Comparison of Sparrows and Owls

RQ1: How do *sparrows* and *owls* differ in terms of participation and quality of contribution? To answer this question we first compared the mean numbers of questions and answers posted by the two groups of users. As depicted



(a) # Answers (b) Reputation (c) Z_{Score}

Figure 2.5: Comparison of expertise metrics.

in Fig. 2.6(a), the ratio between answered and submitted questions is significantly higher for *sparrows*. *Owls*, on the other hand, show a behaviour more similar to average users, thus further highlighting the distinctive "hunger" for answers of *sparrows*.

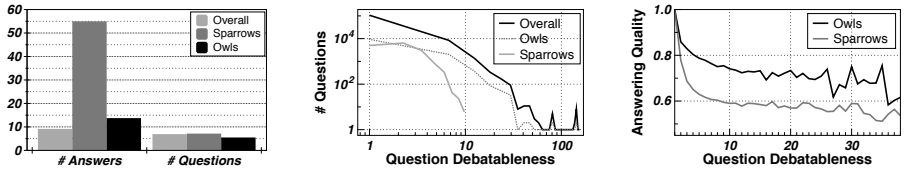
Such a distinction is evident not only in absolute terms, but also with respect to the type of questions and overall utility of answers.

Fig. 2.6(b) shows the distribution of questions answered by *sparrows* and *owls* with respect to the their debatableness: *sparrows* are more focused on questions in a smaller range (and value) of debatableness, while *owls* exhibit a broader range of participation, and a distribution very similar to the one of average users.

Fig. 2.6(c) compares the quality of the answers provided by *sparrows* and *owls* with respect to the debatableness of the answered question. To provide a fair comparison, we just consider questions answered by at least one user in each group. Vertical axis depicts the value of $1-$ *relative ranking* (i.e., $1 - 1/(\mathcal{AU}(u, q_i) * \mathcal{D}(q_i))$). As question debatableness is same for *owls* and *sparrows*, the answering quaity is only determined by utility: a higher value in this figure indicates higher answering quality. We observe that *Owls* consistently provide answers with higher utility, thus showing their grater value for the platform in terms of knowledge creation. The results shown in Fig. 2.6(c) indicate the ability of `MEC` to identify highly valuable users that, even if not driven by the need for higher reputation in the platform, are able to provide relevant and useful answers.

2.4.1 Preferences in Knowledge Creation

This section describes the different behaviours of *sparrow* and *owls* in terms of knowledge creation. We focus on the properties of the questions answered and posted by the two group of users.

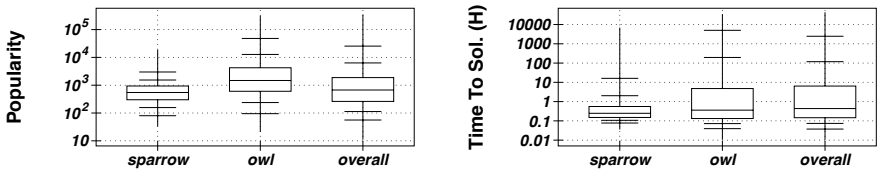


(a) (b) (c)

Figure 2.6: Comparison of activity profiles of *sparrows* and *owls*: a) distribution of number of questions and answers; b) distribution of preferences for question debatableness; c) distribution of quality of contribution for question debatableness.

Finding 1: *Owls* answer questions that are more difficult, and more popular.

We consider two dimensions: **question popularity**, measured in terms of the number of times a question has been viewed in StackOverflow; and **time to solution** [89], measured in terms of the number of hours needed for the question creator to accept an answer as satisfactory. Time to solution can also be an indicator of the difficulty of a question: intuitively, the longer the time to accept an answer, the more difficult is the question.



(a) Popularity (b) Time To Solution (Hours)

Figure 2.7: Comparison of question preferences of *sparrows* and *owls*.

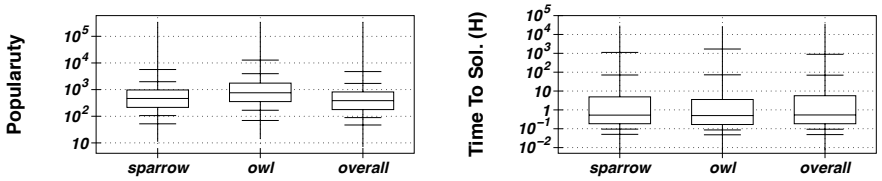
Fig. 2.7(a) shows that questions answered by *sparrows* are, on average, significantly less popular than the ones picked by *owls*. Such difference is even more evident when considering the time required to close a questions – Fig. 2.7(b).

These results might be interpreted as a clear indication of the different motivation and expertise level of the two group of users. *Sparrows* appear focused in building their reputation, which they increase by consistently answering to a lot of easy and non-interesting questions. Their behaviour is however providing important contribution to the community, as they can guarantee fast answers to many questions. On the other hand, *owls* intervene when their expertise is needed the most, i.e. in difficult question. Notice

that such questions are not necessarily the most debated ones, as shown in Fig. 2.6(b).

Finding 2: *Owls* post questions that are more difficult, and more popular.

An analysis performed on the popularity of question posted by *sparrows* and *owls* show another difference between the two groups: questions submitted by *sparrows* are less popular than those posted by the *owls*. On the other hand, the time to completion for such questions is comparable. These results also suggest a difference in the expertise level of the two groups of users, as more popular questions might be a sign of the better understanding that *owls* possess on the subject. However, the higher (on average) difficulty and popularity of *sparrows*'s answers w.r.t. the average of users, also suggests that *sparrows* are good contributors in terms of new problems to be addressed by the community.



(a) Popularity (b) Time To Solution (Hours)

Figure 2.8: Comparison of question posted by *sparrows* and *owls*.

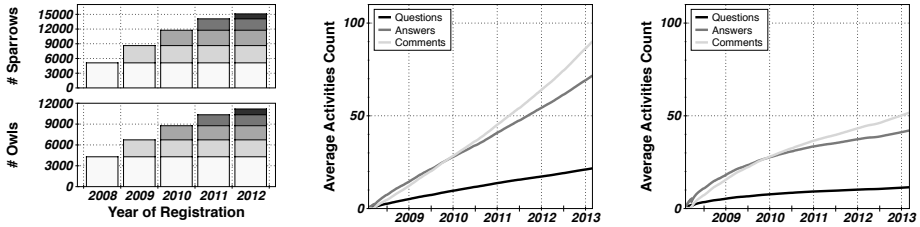
2.4.2 Temporal Evolution of Activities

RQ2: How do the overall activities of *sparrows* and *owls* evolve over time?

Fig. 2.9(a) shows, cumulatively, the number of *sparrows* and *owls* active with the C# topic that registered in StackOverflow. Interestingly, only half of the users in those two categories registered in the first half of StackOverflow's lifetime. A decline can be observed in the number of new registration starting from 2012.

Fig. 2.9(b) and Fig. 2.9(c) describe the temporal evolution of the activities of *sparrows* and *owls*. For each type of users, we extract the number of actions including posting questions, answers and comments, which we refer to the *activity counts*, together with the corresponding timestamp. For each action and for each user group, we averaged the overall amount of activities

in the reference timeframe with respect to the number of *sparrows* and *owls* registered up to that time, plotting the resulting value over the time axis.



(a) (b) Sparrows (c) Owls

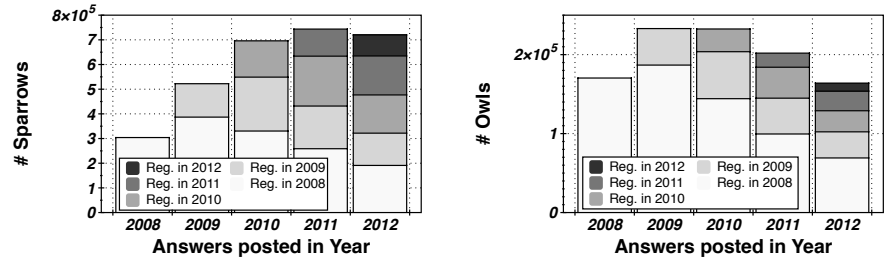
Figure 2.9: Activity evolution of the *sparrows* and *owls*: a) registration date distribution; b) and c) answers, questions and comments.

Finding 3: gamification incentives can more effectively retain *sparrows* than *owls*.

Despite the increasing number of *sparrows* and *owls* over time, the average number of questions per user remains roughly the same, as shown by the black curve in Fig. 2.9(b) and Fig. 2.9(c). This result indicates a relatively stable question posting behaviour, which can be explained in two ways: on one hand, posting questions is not as rewarding (in terms of increased reputation) as providing answers; therefore, what we observe is the result of a genuine question for new information. On the other hand, one can argue that such stable behaviour can be due to a turnover in the number of active users for the topic.

A different behaviours can be observed with answers and comments. The average activity level of *sparrows* increases over time: this is expected, given the important role that reputation incentives play for these users. *Owls*, however, are, on average, less and less active, especially with respect to the number of answers. This result calls for a more detailed analysis of the evolution of *sparrows* and *owls* activities over time.

Fig. 2.10 depicts the temporal distribution of answers given by *sparrows* and *owls* (Figure 2.10(b)) partitioned by the registration date of the answerer. Fig. 2.10(b) shows how "older" *owls* always contribute for the larger portions of the provided answers. However, *owls* consistently tend to decrease their activity in time, especially for more recently registered users. On the other hand, new *sparrows* significantly contribute to a share of answers produced by their group and, although in the long term a decrease in the overall activities of the older member can be seen, the effect is less important. These results



(a) Sparrows (b) Owls

Figure 2.10: Distribution of answers for according to registration date.

suggest that the gamification incentives put in place by StackOverflow are really effective to retain the activity of sparrows.

2.5 Related Work

Collectively edited CQA systems have been emerging as important collective intelligence platforms. A specialized CQA system such as StackOverflow is reforming the way people are communicating and accessing opinions and knowledge [217]. Given such background, matching expertise to the right answerer in Q&A system has recently been a relevant research stream [169, 240, 245]. We introduce the related work by focusing on two aspects: 1) expert finding, and ii) expert modeling in CQA systems.

Expert finding, a classic problem in information retrieval, has been recently re-investigated in the case of CQA systems. An early work [240] focused on the Java developer platform, where it emerged that such expertise network shows a few different characteristics with traditional social networks. In particular, it was found that a simple expertise metric called Z_{Score} (introduced in Section 3) outperforms graph-based metric such as the expertise propagation method (adapted from PageRank). Graph-based methods were then explored for Yahoo! Answers, a much larger CQA platform [112]. A similar topic was also studied in [26], where the author proposed to use the number of best answerers for user expertise estimation. They employed Bayesian Information Criterion and Expectation-Maximization to automatically select the right number of users as experts.

A more recent work [169] adapted Z_{Score} for expert finding in StackOverflow, by using the number of answers a user posted as the ground truth for expertise identification. A similar expertise metric *reputation*, which is

highly correlated with the number of answers, was also used for expert identification in the most recent studies of StackOverflow [89, 168]. However, both metrics are biased to user activeness, therefore partially suitable for StackOverflow due to its gamification design, given that users activities are largely influenced by the reputation and badge rewarding [10]. An important difference between our method for expertise judgement and existing methods is that we take into account the user activeness and eliminate its effect on expertise judgement.

From the point of view of expert modeling, previous works were mostly investigated in the area of software engineering, through analyzing source code [142], version history [113], and developers' interaction history with development environment [69]. Specific to CQA systems, expert modeling focused on modeling the property of questions and answers. In Yahoo! Answers [21], it was found that considering the mutual reinforcing effect between CQA quality and user reputation can improve the effectiveness of expert modeling. Question selection preferences of active users were studied in StackOverflow [169, 165]. While these studies are biased to active user, we target modeling user expertise directly. Our study address the difference between active users and the experts, although the application of our findings is left to future work.

2.6 Conclusion

As CQA systems grow in popularity and adoption, identifying and motivating the users that effectively contribute to their success is becoming more and more crucial. This chapter contributes a novel metric for the characterization of experts in CQA systems, showing its resilience to bias introduced by gamification incentives. Using StackOverflow as reference platform, we investigated differences in the behaviour of most active users (the *sparrows*) and most savvy users (the *owls*), showing how the two groups exhibit very distinct fingerprints in terms of knowledge creation, community participation, and temporal evolution of activities. Although targeted at a single topic, investigations show that similar results can be observed for other topics of similar overall amount of participation.

Chapter 3

Cross-platform Expertise Characterization

Based on our previous results, we extend our study to cross-platforms expertise characterization, so as to capture the multifaceted nature of expertise. We build a dataset linking user profiles in CQA systems with related platforms featuring different user activities. Based on this dataset, we present a principled characterization of two expertise traits, namely, *specialist expertise* and *ubiquitous expertise*, considering their manifestations in both individual and social activities. We further demonstrate the benefit of our expertise characterization approach in question routing.

3.1 Introduction

Expertise is a property of an individual, or a community of individuals that affects the reliability and quality of performance [47] in a given domain of knowledge or practice. Expertise in not created in vacuum: it is the result of the interactions between people with interest in a given domain, where knowledge is created, transferred, and improved within communities. Experts are commonly perceived as those users that can provide an appropriate and correct answer to given a question; or that are able to perform in a correct and timely fashion a given task.

Sociologists have extensively studied the relationship between *expertise* and *expert behaviour* in specialist communities, moving towards a competence-based model of expertise [48]. Expertise is surprisingly difficult to describe, especially because of its diversity in manifestation and representation [85, 220, 226, 229, 237]. In a recent work by Collins and Evans [48], the authors propose the *"Periodic Table of Expertise"* as an attempt to provide a conceptual framework for the organization of different natures of expertise. In their classification, the tacit knowledge expressing domain-specific expertise can be of two main types: i) *ubiquitous*, i.e. knowledge that comes from primary literature (e.g. textbooks, manuals, or the Web); and ii) *specialist*, i.e. knowledge that comes from the process of enculturation in a discipline and that allows its holders to "contribute to the domain to which the expertise pertains". For Collins and Evans, it is only through common practice with others that tacit knowledge can be understood.

The Web is a large scale socio-technical system of resources and people. As such, it provides a unique opportunity to study (at scale) expertise from the perspectives of both individuals and communities. On the Web, specialist communities produce and share resources, and engage in interactional dynamics that are enabled by the social networking features of the adopted on-line platforms.

The measurement of specialist expertise requires to: i) judge one's abilities beyond the mastery of the *language* of a specialist domain [48] (intuitively, being considered an expert may have little to do with one's ability to contribute to the body of knowledge of a discipline); and ii) observe one's behaviour in the context of a community of knowledge and practice. To this end, we advocate the need for a holistic approach in the observation and characterization of expertise, which considers the activities and interactions performed *across multiple systems*.

This chapter focuses on expertise traits and expert behaviour on the domain of knowledge related to the art and craft of *software development*. This choice is motivated by the prominent role played by the Web in shaping and growing programmers. Nowadays, software development is the archetypal Web-mediated profession. Developers are used to: seek information, or provide answers to software-related questions, in question-answering systems; collaborate in software development by using on-line distributed version control systems; and share software-related content on general-purpose social media.

We define an approach for the measurement and comparison of expertise – being it reflected by ubiquitous or specialist knowledge – in three on-line social networks where users perform professional-related activities: StackOverflow, GitHub, and Twitter. We explore an orthogonal axis of analysis, which concerns with the social manifestation of expertise behaviour. We analyse the triggering reasons of one's activities and interactions, by distinguishing between expertise evidences created as the result of an inner drive from those triggered by external stimuli of relational nature. Then, we explore the structural characteristics of the communities discovered in different networks, and study differences in the relative importance of their members. To demonstrate the benefit of a principled characterization of expertise along the above dimensions, we address the problem of improving knowledge creation processes in collaborative question-answering systems.

Our work addresses the following main research questions:

- **RQ1**: Do ubiquitous and specialist knowledge manifest differently across social networks?

  We map user actions on the three targeted networks to types of tacit knowledge, study their occurrence across networks, and investigate their distribution according to the related triggering stimuli (i.e. *individual* or *relational* actions).

- **RQ2**: How can expertise play a role in the formation and activities of communities within- and across-networks?

  By examining cross-platform networks of users on a per-community basis, we aim to understand whether the community structures of the various networks are consistent with each other, and how their composition relates with the knowledge of their members.

- **RQ3**: How can expertise characterization help improve knowledge creation processes in collaborative question-answering systems?

We investigate the problem of question routing, which aims at actively routing questions to the potential answerers to effectively accelerate question-answering processes. We aim to understand how expertise characterized along different types, and different triggering stimuli, could affect the performance of a question routing system.

Contributions. We create and offer to the community a cross-platform dataset of software developer profiles and relations built on top of StackOverflow, GitHub, and Twitter. The dataset links 58K users' accounts across the three platforms, and it includes social networks inferred from the *following* relations in Twitter and GitHub, and the *helper-helpee* network inferred from StackOverflow activities. We provide several original insights about users' expertise manifestation and social expert behaviour within and across networks.

In an extensive set of question routing experiments we show that different types of user expertise, and user actions triggered by different stimuli, could improve the accuracy of finding different types of answerers – i.e. users that provide the best answer, and those that contribute to discussions.

By identifying and studying the activities of individuals and communities active in multiple Web platforms, we aim to push forward research on the per-se challenging problem of expertise characterization in on-line social environments. In this respect, being able to probe not only one's embodiment of relevant skills, but also the socialization into the relevant group practices can provide a vantage point. A better understanding of how expertise and expert behaviour appear on the Web can be used to enable a variety of applications related to professional-oriented services, and to improve the performance of other expertise-driven systems.

The remainder of the chapter is organized as follows. Section 3.2 introduces the dimensions of expertise that are considered in our work. Section 3.3 describes the properties of the dataset created by linking users across StackOverflow, GitHub and Twitter. Section 3.4 analyses expertise and its manifestation in the three networks. Section 3.5 shows how richer expertise characterization can improve the performance of question routing systems. Section 3.6 describes related work, and finally Section 3.7 presents our conclusions.

3.2 Modeling Expertise

This work considers the domain of knowledge related to *software development*, a representative example of profession where expertise can assume multiple forms, like: the sharing of knowledge related to software systems and languages; or the demonstration of actual mastery by creating new software artefacts (i.e. code). Software-related communities are built across multiple Web systems. For instance, developers are used to seek information or provide answers to software development related questions in StackOverflow; to collaborate code on GitHub; and to share software related content through microblogging activities on Twitter. These three platforms are the subject of our study. We here provide a short summary of their properties.

StackOverflow is the most popular question-answering platforms for software developers.[1] Users access StackOverflow to look for existing solutions to their issues, or to post new questions to be answered by topically-defined communities. Platform members can contribute to ongoing discussions with an answer, a comment to a previous contribution, or a preference judgement. Questions, answers, and comments can contain arbitrary text, but often include code snippets, and links to external resources. The platform provides several features (e.g. reputation score, badges) aimed at engaging users, and recognising their contributions.

GitHub is one of the most popular on-line code repository and distributed version control systems.[2] Developers collaboratively develop code hosted in public and private code repositories. Using a "fork & pull" model, users can create their own copy of a repository and submit a pull request when they want the project maintainer to pull their changes into the main branch [114]. GitHub includes social networking features, e.g. to "follow" users and "watch" projects. A personalized profile page can contain identifying information, and a summary of recent activities.

Twitter is a general-purpose microblogging service, where users with different degree of expertise create and share knowledge about a variety of knowledge domains. Our study considers only Twitter content and user interactions relevant to software development and software engineering.

[1] 2.7B yearly global visits during 03/2016 and 03/2017. Source https://www.quantcast.com/, accessed in April 2017.

[2] 21M users and 56M repositories. Source: https://github.com/about, accessed in April 2017.

3.2.1 Ubiquitous and Specialist Knowledge

We borrow the classification of knowledge defined in Collins and Evans' periodic table of expertise [48], which includes *ubiquitous* tacit knowledge and *specialist* tacit knowledge. The former refers to the kind of knowledge that comes with reading primary or quasi-primary literature, i.e. books, manuals, guides, etc. The possession of *ubiquitous* knowledge might require the interaction with "hard", domain-specific material, thus giving the impression of technical mastery.

Specialist knowledge "can only be mastered through enculturation", and it is the only type of knowledge that allows its holders to "contribute to the domain to which the expertise pertains". Simplifying, people with ubiquitous knowledge can *talk* about a subject matter, but only people with specialist knowledge have the ability to *do* things.

These interpretations of *specialist* knowledge put a strong emphasis on the nature of the *artefacts* (e.g., literature, products) that are used or produced to get acquainted with a domain of knowledge, gain fluency and, eventually achieve actionable mastery. They are the reflection of an interpretation of expertise that complies with a competence-based model, depending on "what one can do", rather than "what one can calculate or learn". Also, there is a strong emphasis on the notion of enculturation, which highlights the importance of being an active part of communities of practice in order to acquire understanding of tacit knowledge.

3.2.2 Mapping User Actions to Knowledge Types and Triggering Stimuli

We operationalize the concepts of ubiquitous and specialist knowledge in a set of user activities performed in the targeted platforms. Assuming the ability to identify a user across platforms, by aggregating cross-platform data we can gain a better observation point on the manifestation of knowledge in topics related to software and software development.

Table 3.1 summarizes the mapping we specified for user actions in the three targeted platforms, organized according to i) the type of knowledge; and ii) the triggering stimuli. In the following we elaborate on each of the two categorization dimensions.

Knowledge Types. Actions are classified as *specialist* when they refer to **actionable knowledge**, i.e., actions or content that reflects evidence of

Twitter		StackOverflow		GitHub	
Specialist	Ubiquitous	Specialist	Ubiquitous	Specialist	Ubiquitous
Individual					
■ *AK* tweet	■ Topic Related Tweet	■ Question	N/A	■ *AK own* project ■ *Own* gists	N/A
Relational (within-network)					
■ *AK* RTw.	■ Topic Related RTw.	■ *AK* Answer ■ *AK* Comment	■ Answer ■ Comment	■ *AK others'* project	N/A
Relational (across-network)					
■ Link to *own AK*	■ Link to *others'* content	■ Link to *own AK*	■ Link to *others'* content	N/A	N/A

Table 3.1: Mapping of user actions to types of knowledge. (*AK* stands for "Actionable Knowledge"; *Tw* stands for "Tweet", *RT* stands for "ReTweet").

practical competence. For instance: source code shared on GitHub as part of a project (own or someone else's); code snippets on platforms like *Gist*[3], *Pastebin*[4], or *snipt*[5]; code snippets contained in answers and comments in StackOverflow. This category also includes original tweets in Twitter that i) are related to software development topics and ii) refer to actionable knowledge.

We consider questions on StackOverflow also a manifestation of *specialist* knowledge, and not a lack thereof. Questions reflect an active attempt to acquire actionable knowledge; and they are moderated[6], thus bringing to the community new issues to discuss, and implicitly broadening the available body of knowledge. All other types of actions are marked as *ubiquitous*: given the specialized nature of StackOverflow, we assume answering and commenting to be the reflection of one's fluency with the knowledge domain, although no evidence of practical competence can be directly inferred. Other interaction elements in GitHub (e.g. commit messages) are not considered in this work, and the analysis of their relevance for expertise characterization is left to future work.

Triggering Stimuli and the Social Dimension of Expertise. Expertise is not created in vacuum. It is the result of an interactive process that involves users as both *individuals* and as *members of communities* of people. To

[3] *Gist* are code snippets to be shared with others users. https://gist.github.com/

[4] http://pastebin.com

[5] https://snipt.net

[6] Questions can be marked as off-topic, duplicate. The platform has an automated filter in place that ban questions from accounts that have contributed many low-quality questions in the past. http://stackoverflow.com/help/question-bans

socially characterize expertise, we consider an additional axis of classification regarding the triggering reasons for user activities and interactions. More specifically, we distinguish *Individual* activities performed as the result of an inner drive, from *Relational* actions, which are triggered by community-driven events.

We qualify as individual actions the posting of new topically related Twitter messages, the creation of new questions in StackOverflow, and the creation of new code repositories and sharing artefacts on GitHub.

Relational actions are central to the activities in on-line communities. They are performed to suit a variety of purposes: to communicate, to offer help, or to contribute to the development of their own expertise. For instance, we distinguish between tweets independently created by a user (individual dimension) from tweets generated as response to others' tweets (relational dimension). Relational actions further specializes into *within-platform* and *across-platform* actions, i.e., activities that respectively concern artefacts produced inside or outside the current platform. Content produced in Twitter and StackOverflow might also include reference to external resources, which might be owned by the same content creator. We focus only on content hosted on the three analysed platforms.

Another angle of analysis relates with the *importance* that a user has in a community. To this end, we take into account different indicators of importance expressed at several levels of granularity, including how a user is "voted" by the community members in terms of number of followers in Twitter or GitHub, or through the reputation score obtained in StackOverflow when other users up-vote or down-vote a question or answer.

We also evaluate the user's centrality according to structural properties of a social network inferred from each of the platforms under consideration. More specifically, we analyse followships in Twitter and GitHub, while for StackOverflow we model user relations that express "who helps whom", based on its question-answering system. We shall provide details of the various networks in the next section.

3.3 Evaluation Dataset

This section describes our methodology of creation of professional-oriented cross-platform dataset. We used the April 2015 StackOverflow data dump

available for download from the Internet Archive[7]; and a crawl from the GHTorrent project[8], containing information about GitHub repositories and users updated to March 2015. The working dataset contains 4,132,407 unique StackOverflow user profiles, 4,288,132 GitHub user profiles. Data from Twitter were crawled using its REST API[9] and search functionalities.

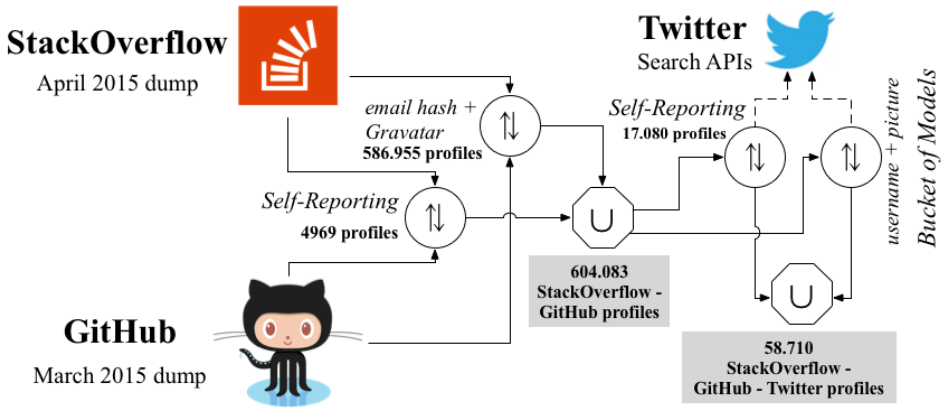


Figure 3.1: Cross-platform profile linking workflow.

To link user profiles across the three platforms, we adopted the methodology presented in [201] and depicted in Figure 3.1. The methodology relies on multiple linking strategies, based on: i) *self-reported links*, i.e. users explicitly mentioning their profiles; ii) users' email MD5 hashes, and `Gravatar` profiles; and iii) fuzzy matching bases on username and profiles pictures. The result is a dataset linking 604,083 StackOverflow profiles with their respective GitHub ones, and 58,710 accounts featuring a link to a Twitter profile. These 58K users and their respective StackOverflow, GitHub, and Twitter activities compose our evaluation dataset.[10]

Web Resources Annotation. Our study builds upon calculated properties of Web resources, i.e. metadata that is not explicitly available in the raw data. We pre-processed Tweets, GitHub pages, StackOverflow questions and answers, and user profiles in the networks to: i) classify resources containing *actionable knowledge*; and ii) classify *link* to external resources, to identify link to content created by the same user on different platforms.

[7]https://archive.org/details/stackexchange
[8]http://ghtorrent.org, accessed at April, 2015
[9]https://dev.twitter.com/rest/public
[10]The dataset will be made available to the community at http://www.wis.ewi.tudelft.nl/ISJ2017

The annotation of actionable knowledge was performed by locating snippets of code in StackOverflow and Twitter content, or by interpreting the type of file(s) contained in GitHub repositories and code snippets. Links included in resources were parsed and analysed to find references to users and other resources contained in the dataset; this allowed us to identify the owner of a StackOverflow, GitHub, or Twitter resource mentioned in any content in the dataset, thus highlighting activities taking place across platforms.

Network Construction. To enable community analysis, we built directed network graphs for GitHub, Twitter, and StackOverflow, in order to model the different user relationships represented in our cross-platform dataset. From GitHub and Twitter, we constructed the graphs denoted as $G_{GH} = (V_{GH}, E_{GH}), G_{TW} = (V_{TW}, E_{TW})$ that encode following relationships of users in GitHub and Twitter, respectively, i.e., a directed edge $e = u \rightarrow v$ indicates that user u follows user v. While being absent of explicit followships, StackOverflow provides an implicit "help network" among users according to *who answers to whom* relationship. Therefore, we constructed a graph $G_{SO} = (V_{SO}, E_{SO})$ such that an edge $e = u \rightarrow v \in E_{SO}$ indicates that user u is helped by v, i.e., at least one question of u is answered by v. This resulted in $37,708$ nodes and $221,780$ edges for G_{GH}, $51,115$ nodes and $1,210,508$ edges for G_{TW}, and $22,903$ nodes and $81,824$ edges for G_{SO}. Noteworthy is the lower density of G_{SO} and G_{GH} with respect to G_{TW}. Networks share the following numbers of nodes: $35,625$ between GitHub and Twitter; $20,375$ between StackOverflow and Twitter; and $16,124$ between StackOverflow and GitHub.

3.4 A Study of Software Expertise Across Web Platforms

In this section we investigate the properties of users and communities in the evaluation dataset, with respect to the expertise characterization dimensions discussed in Section 3.2.

3.4.1 Manifestation of Specialist and Ubiquitous Knowledge across Social Networks

This part of the study is organized around the social dimension of expertise, according to the considered triggering stimuli, namely: *individual* actions,

relational actions involving resources and/or users *within the same network*; and, *relational* actions that *cross-over* networks.

Individual Actions. Twitter messages are the only individual ubiquitous actions considered in our study. They are also the most popular action in the context of our dataset: 48,920 users (83% of the total) produced 2,373,635 topically related Tweets ($\mu = 48.52$, $\sigma = 67.66$, $M = 26$).[11]

Figure 3.2(a) summarizes the distribution statistics related to specialist actions. Given the limited size of Twitter microposts, no specialist actions (e.g. sharing of software code) were discovered. 94% of the population under study manages at least one GitHub public repository ($\mu = 21.13$, $\sigma = 34.87$, $M = 11$), while only 60% has published at least one `gist` ($\mu = 13$, $\sigma = 59$, $M = 2$). The percentage of users is even lower when considering questions published on StackOverflow (20,612 users; $\mu = 17.76$, $\sigma = 45.39$, $M = 6$ questions per user), although 80% of them (16,589) published code as part of their request for help ($\mu = 11.79$, $\sigma = 29.53$, $M = 4$ questions per user). We then conduct correlational study on the amount of expertise actions of the same type performed across different networks. A Spearman's Rho test[12] (0.006 at $p = .43$) revealed the lack of correlation among specialist activity on StackOverflow (questions) and GitHub (code projects). Similar observations also hold when comparing the StackOverflow questions (with code) with GitHub projects, and StackOverflow questions (with or without code) with GitHub gists. These results suggest that users performing one type of specialist actions in one network are not necessarily engaged with comparable amount of specialist actions in a different network.

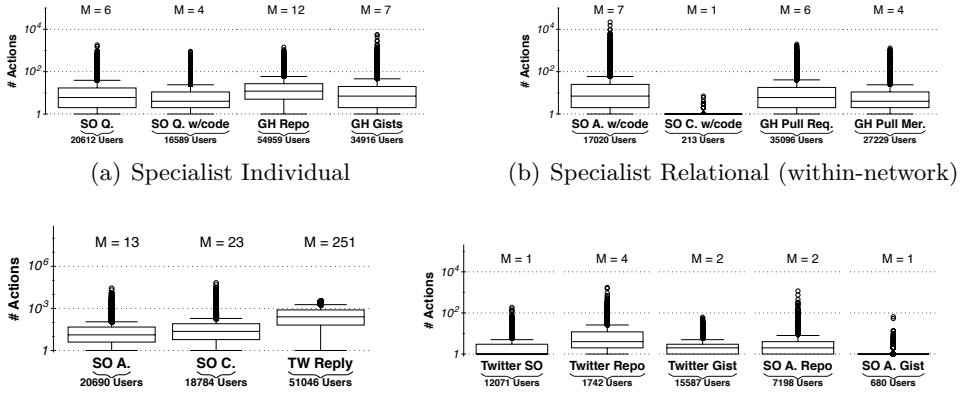
This demonstrates that the manifestations of specialist expertise can vary across networks, thus supporting the need for cross-platform studies.

A comparison of Twitter messages with GitHub code projects and `gists` reveals moderate positive, hence significant correlations (0.214 and 0.249 respectively, both at $p < .01$). These results indicate that users who frequently code in GitHub are also active in posting relevant content in Twitter, thus showing that users performing specialist actions, are also engaged with ubiquitous actions, i.e. the doers are also talkers.

Within-network Relational Actions. Figure 3.2(b) reports the distribution of four specialist actions in StackOverflow and GitHub. 17,020 users (29%) contributed with at least one answer containing actionable knowledge

[11] μ, σ, M stand for the mean, standard deviation, and median, respectively.

[12] Unless differently stated, the properties analysed in the following are to be considered non-normally distributed, thus requiring a non-parametric significance test.



(a) Specialist Individual

(b) Specialist Relational (within-network)

(c) Ubiquitous Relational (within-network)

(d) Ubiquitous Relational (across-network)

Figure 3.2: Summary of the distributions of user actions across networks and expertise types.

(i.e. code; $\mu = 49.60$, $\sigma = 315.96$, $M = 7$), while only 213 (less than 1%) relied on comments as a mean to express their coding capabilities ($\mu = 1.29$, $\sigma = 0.744$, $M = 1$). These numbers assume an interesting dimension when compared to their ubiquitous counterpart (Fig. 3.2(c)), where $20,690$ users (35%) answered at least one question ($\mu = 76.71$, $\sigma = 398.30$, $M = 13$), and $18,784$ users (32%) uploaded at least one comment ($\mu = 159.62$, $\sigma = 994.102$, $M = 23$).

In GitHub, specialist knowledge can be expressed in terms of participation to someone else's coding activity: the majority of users in the dataset ($35,096$ – 60%) requested, at least once, their code to be part of another public repository (*Pull Request*, $\mu = 13.68$, $\sigma = 51.37$, $M = 1$); and a good share of them ($27,229$ – 46%) managed to get their contribution merged (*Pull Merged*, $\mu = 6.54$, $\sigma = 28.50$, $M = 4$). This indicates a good level of specialist interaction *and mastery* within the analysed population.

There is a moderate positive correlation (0.392 Spearman's Rho, $p < .01$) between the action of answering a question is StackOverflow with actionable knowledge, and pull request actions in GitHub; however, the act of simply answering a question (without actionable knowledge) does not correlate with code pull or merge actions (respectively 0.022 and 0.023 Spearman's Rho, $p < .01$). These results suggest that: i) relational specialist and ubiquitous actions are not related to each other; and, more importantly i) the manifestation of relational specialist expertise is consistent over the different networks.

Across-network Relational Actions. Finally, we turn our attention to relational actions that take place across networks. In our setting, these actions take the form of links to content in another platform among those under examination. The dataset contains a good amount of such cross-platform linking, as shown in Figure 3.2(d). In StackOverflow, 7,198 (12%) users included in their answers at least one link to GitHub repositories or *gists* ($\mu = 5.37$, $\sigma = 21.218$, $M = 2$). In Twitter, the number doubles (15,587 users – 26% – with links to `gists`) or quadruples (33,315 users – 57% – link to GitHub repos). This shows that users in the evaluated population show some common attitude in terms of cross-network relational ubiquitous actions. Such sort of expertise manifestation is seldom associated with specialist traits. In our analysis, only 4,713 (8%) users posted links to *their own* repositories or gists in StackOverflow answers, while 1,829 (3%) did the same in Twitter.

Discussion. The study highlights how expertise manifests in many forms, with profession-related ubiquitous knowledge being widely contributed in both Twitter and StackOverflow, whereas more refined forms of knowledge is confined to a relative small subset of users. The amount of activity related to sharing working programs and/or libraries in open repositories greatly varies. Code contribution in GitHub is highly related with code contribution in StackOverflow, both in terms of answers and comments. This indicates that the production of code snippet is a major indication of specialist knowledge in the software domain. Relational actions (answering or commenting someone else's contribution, or re-distributing software related tweets) are relatively common in the population under analysis. A consistent subset of the StackOverflow population is engaged with relational activities. Answers are the preferred way to contribute specialist knowledge to the community, although their frequency is significantly lower.

When analysing cross-platform expertise manifestation, we observe a clear prevalence of ubiquitous knowledge, i.e. links to someone else's content. When producing actionable knowledge on StackOverflow, users prefer to directly provide code, instead of simply suggesting a previous solution published elsewhere. We account this to the competitive nature of a platform like StackOverflow, where gamification techniques steer people behaviour toward actions that maximize reward. On the other hand, users appear to be very selective about the source of referenced actionable knowledge, where only few feel confident enough to share their code as reference. This is an unexpected result, that calls for further investigation: it would be interesting,

for instance, to study the actual quality of the shared code (using, e.g. code analysis or review).

3.4.2 The Role of Expertise in Communities Within- and Across-Networks

This question is addressed by focusing on properties deriving from the networked organization of the targeted platforms.

Within-Network Popularity. Experts are important and active components of communities of knowledge or practice. Therefore, *reputation* and/or *popularity* aspects account as relevant properties of study. First, we analyse the popularity of *individual* users according to metrics explicitly defined in the targeted networks. Then, we rely on network-centrality analysis to identify relevant users and characterize their actions *within* the respective networks.

Figure 3.3 compares the distribution of several profile properties of users in StackOverflow, GitHub, and Twitter. Almost all users in the considered population have at least one social connection in Twitter (89%), and performed at least one rewarded activity in StackOverflow (90%). Social connections in GitHub are slightly less common, with 81% of users having at least one follower, and 73% of users following at least another user. There exists a strong correlation between profile properties of the same network. For instance, the number of *followers* and *followees* in GitHub (0.653 Spearman's Rho, $p < .01$), *score* and *upvotes* and *downvotes* in StackOverflow (0.872 and 0.721 respectively Spearman's Rho, $p < .01$), and number of *followers*, number of *friends*, and *number of tweets* in Twitter (respectively 0.694, 0.674, and 0.511 Spearman's Rho, $p < .01$).

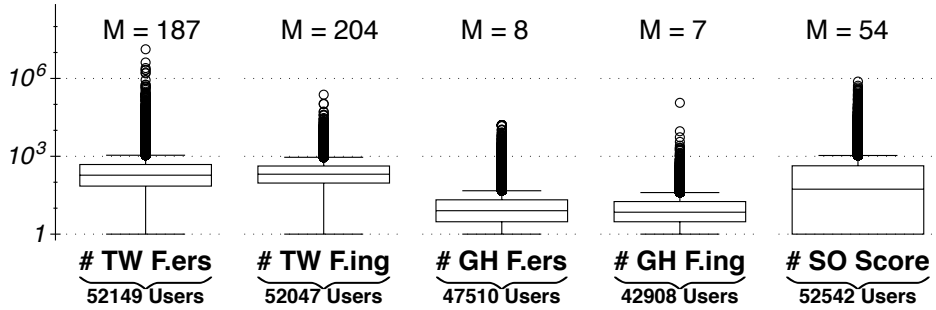
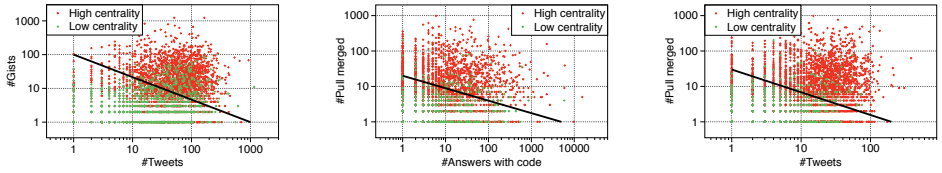


Figure 3.3: Network-related properties in Twitter, GitHub, and StackOverflow. Box plots include only users having values greater than 0.



(a) Individual specialist action Gist vs. ubiquitous action Tweet

(b) Relational specialist action Merge vs. specialist action Answer

(c) Relational specialist action Merge vs. ubiquitous action Tweet

Figure 3.4: Scatter plots that compare the different (specialist/ubiquitous) actions of users of different centrality.

Significant correlation emerges also for profile properties *across networks*. The number of followers in GitHub is positively correlated with StackOverflow reputation (0.237 Spearman's Rho, $p < .01$ with Bonferroni correction), number of followers (0.397 Spearman's Rho, $p < .01$ with Bonferroni correction) and friends (0.213 Spearman's Rho, $p < .01$ with Bonferroni correction) in Twitter. Another positive yet weak correlation exists between *score* in StackOverflow and the number of topically-related tweets in Twitter (0.160 Spearman's Rho, $p < .01$). We observe a significant lack of correlation between StackOverflow *score* and the overall number of topically related tweets (0.063 Spearman's Rho, $p < .01$).

To further understand the impact of within-network popularity on user behaviour, we used the classic PageRank model to calculate users' centrality scores in each networks. We focus the following analysis on centrality scores in GitHub, due to its strong specialist nature. Figure 3.4 depicts the distribution of several types of GitHub actions; to highlight differences, we consider users in *first* and *last* quartiles of the GitHub centrality ranking. Red dots represent users with high centrality score, while green dots represent users with low centrality. The former are more active in terms of `gist` and merged pull requests. The activity of the two groups are clearly distinguishable in other networks. For instance, high centrality users are also very active in Twitter, and often provide answers is StackOverflow that contain code snippets. These results conform with the observation emerged form the analysis of individual actions in Section 3.4.

Analysis of Centrality Across-Networks. We analyse the correlation of users' centrality scores across networks. Intuitively, a high cross-network correlation would indicate similar user importance in different settings; for instance, a high correlation of user centrality in StackOverflow and GitHub networks will suggest that users that are helpful in answering to others' ques-

tions in StackOverflow would be popular (i.e. followed by many users) in GitHub (and vice versa); a low correlation would indicate that users' importance in one platform is not indicative of their relevance in another platform, e.g., an influential user in GitHub may not likely to answer questions in StackOverflow.

We performed two stages of evaluation, using *Kendall-tau rank correlation coefficient* and *Fagin's intersection metric* [67]. Kendall-τ correlation evaluates the similarity between two rankings, expressed as sets of ordered pairs, based on the number of inversions of pairs which are needed to transform one ranking into the other. Kendall-τ correlation is expressed in the interval $[-1, 1]$, where 1 (resp. -1) means that the two rankings are identical (i.e., one reverse of the other). We calculated the correlation of PageRank ranks between every pair of network graphs, which resulted in strong (0.411) correlation between GitHub and Twitter, and moderated (0.141) correlation between GitHub and StackOverflow, and between StackOverflow and Twitter (0.154).

Fagin's intersection metric determines the agreement of two ranking lists by accounting for "partial rankings" (elements in one list may not be present in the other list) and top-weightedness, i.e. the top of the list gets higher weight than the tail. Fagin score is the average over the sum of the weighted overlaps based on the first k nodes in both rankings, and it is in $[0, 1]$, where higher values correspond to better agreement. Figure 3.5 shows the results, which are consistent with the previous correlation analysis: the ranking agreement between Twitter and GitHub is good even for lower values of k, whereas StackOverflow ranking appears to be much less correlated with both the other two networks.

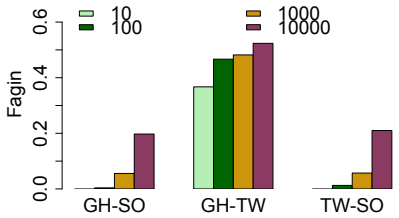


Figure 3.5: Fagin's intersection metric, with top-weightedness parameter $k \in \{10, 100, 1K, 10K\}$. (GH, SO, and TW are used as abbreviations of GitHub, StackOverflow, and Twitter networks.)

Cross-platform Community Structure. We now investigate how users connect with each other, and how they tend to form communities in the different networks. We do so by exploring the structural characteristics of

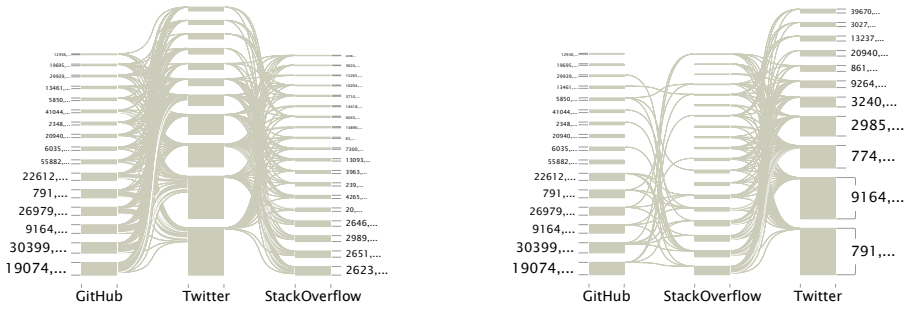


Figure 3.6: User matching between the largest communities of different networks: (a) GitHub-Twitter-StackOverflow and (b) GitHub-StackOverflow-Twitter. Each layer corresponds to the network-specific community structure detected by Infomap. Communities correspond to the modules, whose height is proportional to the community size. The transition curves connect nodes assigned to communities in the different networks. The module label corresponds to the node with largest flow volume in the community. *(Best viewed in the electronic version.)*

the communities discovered in the three networks. For the task of community detection, we used the two-level formulation of the state-of-the-art *Infomap* algorithm [188]. *Infomap* is a search algorithm that minimizes the flow-based *map equation* model, which relies on the principle that communities are detected not from the network topology only, but as groups of nodes among which the flow persists for a long time once entered.

Figure 3.6 shows the cross-platform matching of users on a per-community basis. The transition curves can be interpreted differently depending on the streamline direction. Considering the matching between the GitHub and Twitter followship networks – Figure 3.6(a)), we observe that users of each particular community in Twitter belong to different communities in GitHub; moreover, in the 1st, 2nd and 5th largest Twitter communities, a significant proportion of users are involved in matching with GitHub communities. Focusing on the opposite streamline direction, almost all users in the largest GitHub communities are found as members of Twitter communities, and the transitions from the largest GitHub communities appear to be much more concentrated towards one or few Twitter communities. A similar scenario of one-to-many community alignments is observed between Twitter and StackOverflow, although the flow volume appears to be smaller with respect to the comparison between Twitter and GitHub. Also, there is a tendency for

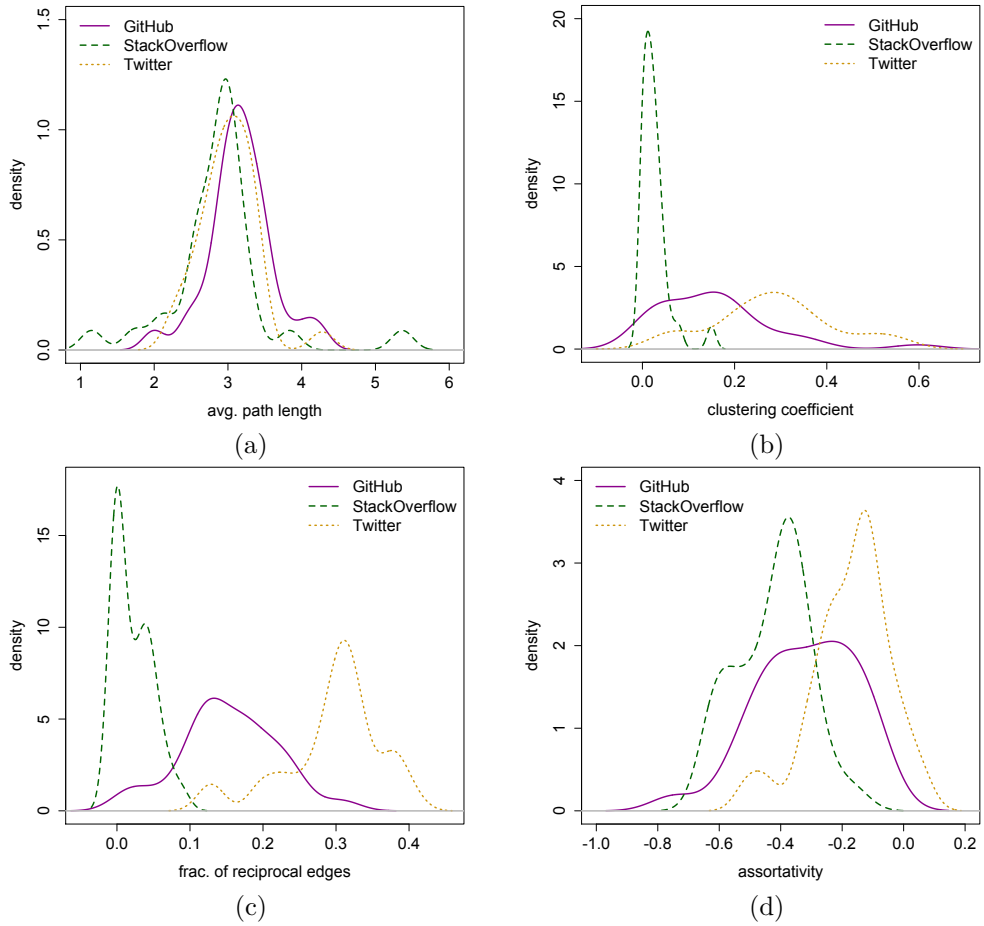


Figure 3.7: Density distributions for top-30 largest communities: (a) avg. path length, (b) clustering coefficient, (c) fraction of reciprocal edges, and (d) assortativity.

users in a StackOverflow community of being members of several Twitter communities. This would indicate a weaker consistency between the community structures of these two networks. Figure 3.6(b) shows how the GitHub and StackOverflow communities are very disconnected from each other, thus suggesting that the two platforms follow different mechanisms of community formation.

Structure Analysis of Communities. We also investigated *small-world hypothesis*, *homophily* and *reciprocity* aspects in selected communities from each of the networks. Figure 3.7 shows density distribution plots corresponding to the induced subgraphs of the top-30 largest communities found on

each platform[13]. The distributions refer to the average path length, clustering coefficient, fraction of reciprocal edges, and assortativity measures. Figure 3.7(a) show that average path length of community is relatively low in all networks. The three density distributions are also quite narrow, meaning that most of the communities have similar average path length (3.2 for GitHub, 2.9 for StackOverflow, and 3.0 for Twitter). Figure 3.7(b) shows the density distributions of clustering coefficient of community. Most communities in StackOverflow tend to have very low probability that "friends of a friend" are connected to each other (3rd quartile equal to 0.034). GitHub and Twitter communities reflect a different scenario, with highest peaks for larger clustering coefficient values (between 0.15 and 0.3), sparseness and, in the case of GitHub, with slight right-skeweness. Considering the above remarks about average path length, we can conclude that, on average, communities in Twitter and, to a smaller extent in GitHub, tend to behave much more likely as a small-world than in StackOverflow.

Concerning reciprocity, Figure 3.7(c) shows the distributions of the fraction of reciprocal edges. These appear to be roughly centred around different regimes, i.e., 0.011 for StackOverflow, 0.15 for GitHub, and 0.31 for Twitter.

That is, reciprocity turns out to be mostly a rare event in StackOverflow, while in GitHub and Twitter on average one every seven and one every three connections, respectively, is reciprocated. Like for the clustering coefficient analysis, GitHub and Twitter communities present a much sparser distribution than in StackOverflow. All three network communities show a tendency towards negative assortativity – Figure 3.7(d) indicates that nodes with similar degree are not likely to connect to each other. StackOverflow presents the lowest level of homophily (peak of -0.40), followed by GitHub (-0.30) and Twitter (-0.15). In contrast to the previous distribution analyses, the trend is less narrow for GitHub than for the other two networks, while nonnegative assortativity is shown only at the 95th percentile.

Discussion. The study unveils a number of insights about the interplay of expertise and the characteristics of online networks, and their communities. The popularity of a user within a network is strongly correlated with the amount of performed actions, a result that confirms conclusions from previous work. We discover the existence of similar correlation also *across networks*, showing how activity and popularity are properties of a category of users that consistently manifest their ability across professional networks.

[13]A community-specific induced subgraph contains edges only between nodes belonging to the community.

The comparison of centrality scores also reveals interesting differences. The most central users on StackOverflow do not have the same importance both in GitHub and Twitter, which we tend to ascribe to the different type of interaction (i.e., "who helps by answering to whom") considered for StackOverflow. By contrast, in the GitHub-Twitter case, it turns out that user centrality in the two fellowship networks are quite correlated, thus suggesting a link between the property of being an "opinion leader" in Twitter and and a popular "coder" in GitHub.

The analysis of communities is consistent with the previous consideration. We observe consistency of structure between the followship networks in GitHub and Twitter, while not in GitHub and StackOverflow communities, which appear to be very disconnected from each other, thus suggesting that the two platforms follow different mechanisms of community formation. Considering topological properties exhibited by the most representative communities in the three networks, one interesting remark is that communities in Twitter and, to a smaller extent in GitHub, tend to behave much more likely as a small-world than in StackOverflow communities. The latter also show less reciprocity among users, which suggests that users who play the role of helper do not tend to ask for help and vice versa.

One common aspect to all networks is the negative tendency of users with similar degree to connect to each other. This result is expected in a *"helper-helpee"* network, and confirms previous studies on followship networks.

3.4.3 Limitations

This section provides two main contributions: an approach to the measurement and comparison of specialist expertise (as reflected by ubiquitous or specialist knowledge), as a property of individuals and communities, within and across online platforms; and a study performed over a domain of knowledge, namely software development. While the former contribution is general, the latter is relevant only for software development, and cannot be directly generalized to other domains of knowledge. However, as discussed in Section 2, software development is a representative example of profession where expertise can assume multiple forms, and where the Web plays a fundamental role of expertise sharing and development. This makes the experimental evaluation described in this chapter potentially useful for supporting similar research targeting the domain of knowledge having similar characteristics, such as those exhibited in collaborative enterprise networks. While a study performed on a larger set of individuals and platforms would have been de-

sirable, finding realistic and sufficiently rich datasets has been challenging. To the best of our knowledge, our dataset of 58K profiles linked across three networks is the biggest to date, in the domain of software development.

3.5 Exploiting Cross-Platform Profiles for Question Routing

Question answering platforms like StackOverflow provide a prototypical use case to demonstrate the need of better expertise assessment on the Web. Given their growing importance, question routing and recommendation can greatly improve the speed and quality of the provided answerers [235]. In this section we answer our **RQ3**, demonstrating how the *question routing* problem can benefit from a principled characterization of expertise. The focus is on the $58,710$ users that compose our dataset of study. Based on their answering history in StackOverflow, we examine the performance effect on question routing of: i) the adoption of a user model that insists on different types of specialist expertise (i.e., ubiquitous and specialist); and ii) the characterization of user actions according to the triggering stimuli (i.e. individual actions, and within- and across-network relational actions).

3.5.1 Experimental Setup

Data preparation. As common in question routing experiments [235], we consider only questions answered by at least three users (i.e. answerers) in our dataset. The filtering resulted in a question-set composed by $19,478$ distinct questions, answered $64,182$ times by a total of $12,272$ distinct users. Noteworthy to the following interpretation of the results is the sparsity of the resulting question-answerer matrix, that is $2.68e^{-4}$.

For each question, answerers are ranked according the number of received votes, thus interpreting the appreciation from the community to the answer as an indication of answering quality. As the distribution of #votes is scale-free, a #votes x associated with an answer is normalized by applying $\frac{1}{1+x^{-1}}$, a normalization scheme that has been proved to be effective in relevant recommendation problems [233].

Configurations. To investigate the benefit of expertise characterization in question routing, we incorporate the measures of the amount of expertise-related user actions across networks. Following the mapping of user actions to specialist knowledge types and triggering stimuli, as classified in Table 3.1,

these measures are aggregated according to the following dimensions: i) **Expertise Type**: where attributes related to ubiquitous and specialist expertise are respectively aggregated; ii) **Triggering Stimuli**: where measures related to individual, within- and across-networks actions are respectively aggregated.

Each aggregation (e.g. aggregation of ubiquitous expertise-related measures) results in a specific user expertise matrix that describes a certain facet of user expertise (e.g. ubiquitous expertise). We then study the additional predictive power of these user expertise matrices by comparing them with the baseline experimental configuration, where the recommendation is solely based on historical question-answering records.

We use the Matrix Factorization (MF) model [124, 159] as the method for baseline configuration. To enable the incorporation of expertise characterization, we extend the MF model by co-factorising both the question-answerer matrix and expertise matrix, as detailed below.

Routing model. Let $\mathcal{Q} = \{q_1, q_2, \ldots, q_m\}$ be the set of m questions, and $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be the set of n users. Given a question-user matrix $\mathbf{V} \in \mathbb{R}^{m \times n}$, we denote with $\mathbf{V}_{ij}$ the (normalized) voting received by user u_j to question q_i. In the case of no-answer given by user u_j to question q_i, we set $\mathbf{V}_{ij} = 0$. We then construct a binary mask matrix by letting $\mathbf{O} \in \mathbb{R}^{m \times n}$, where $\mathbf{O}_{ij} = 1$, indicating that u_j answers to v_j, and $\mathbf{O}_{ij} = 0$ otherwise.

For the baseline routing configuration, we apply the MF model taking as input only the historical interaction between answerers and questions. The key idea of MF is to learn the latent factors of both questions and users from the observed voting entries in $\mathbf{V}$, then use such latent factors to predict unobserved ones. MF solves the following optimization problem:

$$\min_{\mathbf{Q},\mathbf{U}} \quad \frac{1}{2} \sum_{i,j} \mathbf{O}_{ij} (\mathbf{V}_{i,j} - \mathbf{Q}_i \mathbf{U}_j^T)^2 + \frac{\lambda}{2} (\|\mathbf{Q}\|_F^2 + \|\mathbf{U}\|_F^2), \tag{3.1}$$

where $\mathbf{Q} \in \mathbb{R}^{m \times k}$ and $\mathbf{U} \in \mathbb{R}^{n \times k}$ are the latent factor matrices of questions and users to be learnt; k is the dimensionality of the latent factors; $\|\cdot\|_F^2$ is the Frobenius norm of a matrix, to regularize matrix norms of $\mathbf{Q}$ and $\mathbf{U}$ to avoid overfitting; λ is the parameter for controlling regularization strength.

To compare the proposed configurations, we extend the basic MF model to incorporate expertise-related measures. The new model co-factorizes the user expertise matrix, which we denote as $\mathbf{Y}$, with the voting matrix $\mathbf{V} \in \mathbb{R}^{m \times l}$,

formulated as follows:

$$\min_{\mathbf{Q},\mathbf{U},\mathbf{Z}} \frac{1}{2}\sum_{i,j}\mathbf{O}_{ij}(\mathbf{V}_{i,j}-\mathbf{Q}_i\mathbf{U}_j^T)^2+\frac{\alpha}{2}\sum_{j,l}(\mathbf{Y}_{j,l}-\mathbf{U}_j\mathbf{Z}_l^T)^2 + \frac{\lambda}{2}(\|\mathbf{Q}\|_F^2+\|\mathbf{U}\|_F^2+\|\mathbf{Z}\|_F^2), \tag{3.2}$$

where $\mathbf{Z} \in \mathbb{R}^{l\times k}$ is the latent factor representations of expertise measures. The latent user matrix $\mathbf{U}$ is factorized from both $\mathbf{V}$ and $\mathbf{Y}$, thus leveraging both historical user question interaction and expertise information for enhanced latent factor representation of users.

Evaluation. We applied 5-fold cross-validation and report the average performance. In each fold, we masked off 20% answers for test, and used the other 80% for model training. We empirically set optimal parameters using a grid search in $\{0.0001, 0.001, 0.01, 0.1\}$ for both λ and α. The dimensionality of latent factors k is set to 50. Note that we are not interested in predicting #votes, a value that is known to be highly correlated with the popularity of a question [235]. Instead, we focus on the prediction of the users that actually provided an answer to the question. This choice results in a top-n recommendation scheme [60]. For each question, and for all the users actually answering to it but that *do not appear in the training data of that question*, we estimated a value of answering quality, which was then used to rank them for recommendation. The ranking specific to a particular question was then compared to the ground-truth ranked list of answerers to that question, based on which we computed *precision*@N, *recall*@N and $NDCG@N$ where N is the number of recommended answerers [193].

3.5.2 Results

Table 3.2 reports the performance of question routing with different configurations. We observe that both *ubiquitous* and *specialist* expertise can improve question routing performance. A Wilcoxon signed rank test reported differences significant at $p < 0.01$ for all pairwise combinations, except for *precision*@1. Ubiquitous expertise appear to contribute most to performance improvement, especially when $N = \{5, 10\}$. We explain the result with the importance of users' activeness in expert interaction: being able to express and inference domain knowledge, regardless of one's actual operationalization abilities, significantly associate with the ability of helping with topically related questions. Specialist expertise contributed the most with the retrieval of the best candidate ($N = 1$); this result shows that our expertise categoriza-

Evaluating Metric	Baseline Model	Expertise Types		User Actions			Expertise Ensemble
		Specialist	Ubiquitous	Individual	Relational(w.)	Relational(a.)	
Prec.@1	.0524	**.0573**	.0546	**.0560**	.0546	.0545	**.0596**
Prec.@5	.0229	.0292	**.0334**	.0278	.0292	**.0317**	**.0368**
Prec.@10	.0147	.0197	**.0229**	.0188	.0201	**.0212**	**.0245**
Recall@1	.0420	**.0460**	.0436	**.0446**	.0441	.0441	**.0485**
Recall@5	.0900	.1152	**.1327**	.1100	.1152	**.1261**	**.1457**
Recall@10	.1164	.1558	**.1814**	.1480	.1587	**.1685**	**.1936**
NDCG@1	.0439	**.0482**	.0456	**.0467**	.0461	.0459	**.0505**
NDCG@5	.1450	.1976	**.2308**	.1847	.1962	**.2199**	**.2549**
NDCG@10	.2320	.3308	**.3915**	.3106	.3390	**.3576**	**.4129**

Table 3.2: Question routing performance. For each evaluation criterion, the best performance per category of configuration is highlighted in bold.

tion scheme has value also beyond the characterization of expert behaviour, with concrete application to different routing objectives.

Focusing on actions types, we see that they can all contribute to question routing performance improvement. A Wilcoxon signed rank test confirmed statistical significance at $p < 0.01$ for all pairwise combinations. Individual actions achieved the best result in the case $N = 1$, thus indicating their value in retrieving the best answerer to a question. In other cases, within-network and across-network relational actions proved more effective. We explain the result as follows. The average number of answers in StackOverflow is 3: questions with more answers are usually the more debated ones, thus requiring users more able to engage in discussions. By modeling network actions, relational activities are able to bring out such a latent behavioural character of users. Finally, in the case of $N = \{5, 10\}$, across-network user actions shows superior performance to with-in network user actions, thus verifying the benefit of cross-platform expertise modeling.

Comparing with the performance of user expertise measures categorized by trigger stimuli, we could observe that expertise types yield, in general, higher performance. This implies that proper categorization of expertise according to the specialist expertise types could help more in improving question routing performance.

The last column of Table 3.2 reports the performance obtained by ensembling all expertise types, the best in all evaluations. This result is a clear indication of the combined benefit of both expertise types, and serves as a comparison term to assess the performance penalty to be paid with partial expertise quantification.

3.6 Related Work

Expertise modeling is an active research topic in the study of the Web as a social platform. It is relevant to many domains including information retrieval (IR), social network analysis, user modeling, and Software Engineering. Seminal works in IR on expert finding have made great impact in both online [239] and enterprise [15] applications. With the growing pervasiveness of research on the science of the Web, several expertise attributes, and expertise identification methods have been proposed, e.g. user profiles and posted content [221]; user motivations [**?**]; online questions and answers [168], and working activities [69]. Among existing methods, two expertise identification class of techniques emerge as most common: i) metric-based methods, such as answering quantity [168, 240], and quality [229]; and ii) graph-based method, which essentially measures the importance of users in their interaction [26, 112, 240].

This chapter contributes a principled categorization of expertise attribute, that can be applied across domains of expertise and domain-specific platforms. Previous work addressed the study of cross-platform user interaction and influence in general-purpose social networks. These include methods for linking user accounts across networks [45, 162], and for modeling cross-network user interests [2, 3]. Specific to the domain of software development, Valiescu et al. [216] compared StackOverflow and GitHub users' activities, to explore the existence of correlation between participation in StackOverflow and productivity on GitHub. Badashian et al. [14] looked also at influence and recognition that each user has within and across these two platform. Buccafurri et al. [34] linked users profiles to investigate to which extent users favour the information flow across-networks.

This chapter relates with these previous efforts, but radically depart from them as we: i) formally characterize different types of expertise; ii) include an additional platform of analysis; iii) significantly extend the set of studied users; iv) provide network characterization of expertise. Thanks to these original contributions, we advance the state of the art in the field, and provide a benchmark for future studies in cross-platform expertise characterization and application.

Question routing is an important technique for improving question answering process [**?**]. Previous work typically considers only user activeness as approximation of expertise for question routing [84, 170, 40]. Our work differs from them by extensively studying the effect of different types of expertise on question routing performance. This chapter is also related to the growing

body of research on cross-domain recommender systems [19, 194, 191], which in general considers data from multiple domains to enrich rating-based user models in the target domain. This work complements existing research by showing how user expertise principally characterized in multiple dimensions could enhance the question routing performance.

3.7 Conclusion

In this chapter we pushed forward research on analysis of expertise and its manifestations on Web-based networked environments. Our work has stressed the importance of being able to identify different classes of expertise traits and user actions, and the need of cross-network analysis to better capture the different manifestations of expert behaviour.

Using the software development domain as case in point, we described the measurement and comparison of specialist expertise in three professional-oriented online platforms: StackOverflow, Twitter, and GitHub. By capitalising on a dataset of 58K linked user profiles (the most extended to date), we determined how individual and relational actions vary across networks and communities, providing a number of original insights. For instance, we have shown that a small set of users manifest *specialist* knowledge, whereas such manifestation is consistent in both GitHub and StackOverflow. Our cross-platform analysis showed that active users are popular *across* networks, and that top coders in GitHub tend to also be opinion leaders in Twitter. The benefits of our principled approach to expertise modeling are demonstrated by the improved performance of question routing in StackOverflow.

Chapter 4

The Social Dimension of On-line Microwork Markets

To fully exploit crowds for knowledge creation acceleration, this chapter provides a study with a complementary perspective w.r.t. the previous chapters, namely, the relationships between crowd preferences and knowledge creation demand and outcomes. We therefore analyze crowd discussions in fora, and task availability and executions in marketplaces. As the result, we present sufficient evidences to show the mutual influence between crowd discussions and market dynamics.

4.1 Introduction

Micro-task crowdsourcing has the power to reach to large amounts of individuals for work execution. As such, it has become a very appealing approach for data collection and augmentation purposes, and platforms such as Amazon Mechanical Turk (mTurk) and Crowdflower are still on the rise.

Microwork markets are socio-technical systems, regulated by complex mechanisms that relate the activities of requesters and crowdworkers. This class of online labour has been widely studied in many aspects, from crowdworker analysis [30, 61], to market analysis [104, 63]; from incentive mechanism design (e.g. pricing schemes) [77], to crowdworker retention [62]. However, previous research is characterized by a common yet faulty assumption: crowdworkers are anonymous, and their activities occur in isolation, oblivious of factors external to the market.

Recent work has challenged such assumption [82, 148], showing that crowdworkers interact and collaborate outside microwork markets, in online fora such as `mTurkForum` and `Turkopticon`. These fora are virtual social environments that aim at developing social capital in online labour, by supporting the social and technical needs of their members. Using survey and forum data, Yin et al. [238] discovered that a high proportion of crowdworkers use at least one forum (59.1% of all crowdworkers involved in their survey), and that crowdworkers within the same forum are more likely to establish direct interactions with their fellows. These findings clearly point to the need for a better understanding of crowdworker communities and their relationships with microwork markets.

In this chapter we investigate the mutual influence between crowdworker community activities in online fora and the mTurk microwork market. We hypothesize that the activities of crowdworkers in fora are influenced by – and can influence – the status of mTurk. Specifically, we investigate: **1)** how discussions in crowdworker fora relate with the properties of published HIT groups, with the temporal distribution of HIT groups availability, and with the properties of the publishers (i.e. requesters) of such HIT groups. And, **2)** the relationship between tasks that are discussed in crowdworker fora, and their execution speed in mTurk.We seek answer to the following research questions:

- **RQ1**: How is the activity of crowdworker communities in online fora influenced by the content, requesters, and variations of availability of HIT groups in the mTurk market?

- **RQ2**: To what extent does the activity of crowdworker communities affect task consumption in the mTurk market?

We address these questions by collecting, enriching, and analysing a dataset of discussions produced by crowdworker communities in six online fora[1] in 6 years. We also retrieve information about more than 2.6M HIT groups published in the Amazon Mechanical Turk market, including data about their publication and completion over time. We link these HITs with related messages in fora, and use time series analysis techniques to highlight the correlation between activities on the market and activities in the fora.

Crowdworker fora provide a unique vantage point to observe the activities of a large amount of crowdworkers. Differently from previous studies that analysed crowdworkers by means of ad-hoc tasks published in the mTurk market [153, 187, 238] (and thus subject to inherent selection bias), our broader, outside-in analysis provides novel insights on the behaviour and work-related preferences of crowdworkers organized in online communities. This chapter provides the following original contributions:

- A novel dataset, linking 6-years worth of mTurk-related fora discussions. The dataset contains 3.1M messages, produced by 28.9K members. Messages are classified according to their content type (e.g. Comment, Experience, Social), and are linked to 184K distinct HIT groups and 51K distinct requesters.
- An analysis of the relationship between properties of HIT groups in the market, and their likelihood of being mentioned in fora. We show that the amount of tasks of a given type (e.g. Survey, Content Creation) available in the market is not predictive for the amount of discussion about that task type in the fora.
- An analysis of the relationship between HITs availability in the mTurk market, and discussions in fora. We found significant synchronicity between the two time series, with an average positive lag of 4 hours, and shortest positive lag of 45 minutes. We show that the temporal distribution of HITs availability can help in the prediction of crowdworker discussions, with significant differences across fora and discussion categories.

[1]Namely `mTurkCrowd`, `mTurkForum`, `TurkerNation`, `mTurkGrind`, `Reddit HWTF`, and `Turkopticon`.

- An analysis of the relationship between properties of requesters, and their likelihood of being mentioned in fora. We show that requesters with higher communicativity and generosity are more likely to be mentioned in all fora. Fairness and promptness, on the other hand, have a significant effect only in few fora.
- An analysis of the relationship between discussions in fora and the consumption rate (throughput) of HIT groups in the mTurk market. We found quantitative evidence of the positive effect that HIT groups' mentions in fora have on HIT consumption throughput (on average, a 59% improvement in the first hour). A targeted analysis of temporal synchronicity show that the temporal progression of mentions in fora can help in the prediction of throughput, with an average positive lag of 30 minutes, and an average 340% boosting effect.

A deeper knowledge about the relationship between crowdworkers, crowdworker communities, and microtask markets is of crucial importance for a variety of purposes, including but not limited to the design of tasks, incentive and task allocation schemes, and human computation systems [73, 227].

4.2 Related Work

The focus of our work is on the Amazon Mechanical Turk market (mTurk). In mTurk, *Requesters* provide work in the form of HITs (Human Intelligence Tasks) organized in groups; *Crowdworkers* actively seek for HIT groups to complete and to be rewarded for. The platform allows access to crowdworkers residing in US and India [104]. Fort et al. [68] estimated active crowdworkers to be between 15 and 43 thousands, with most of the tasks (80%) carried out by the most active workers (20%, between 3000 and 8000).

Previous work investigated the complex mechanisms that regulate microwork markets, with the goal of understanding their properties and the behaviour of their actors. Most notably, [63] provides a long-term analysis of the mTurk market, showing that the size and recency of HIT groups are two key features for the prediction of the completion time (throughput) of a HIT group. Little work focused on the analysis of the impact of crowdworker on mTurk.

Crowdworker Communities. Studies of work and occupations are most fruitful when they are grounded in crowdworkers' experience, rather than narrowly focusing on the macro or micro mechanics of economic productivity

and efficiencies [1]. In this work, we embrace this perspective and analyse the mTurk market from the point of view of crowdworker communities that operate in online discussion fora. Only recently, researchers have started to investigate online crowdworker communities [82, 126, 238]. In the context of the mTurk market, crowdworkers organize in communities around a number of online discussion services [82, 105, 238] such as `Turkopticon`, `mTurkCrowd`, and `TurkerNation`. As true in many online fora, members of crowd communities turn to these services to serve three fundamental needs for their members: functional needs (e.g. skill improvement, information sharing), social needs (e.g. building community and trust between workers, providing collective protection), and psychological needs (e.g. providing moral support and encouragement to each other) [82, 126].

The resulting socio-technical system is sketched in Figure 4.1, where the microwork market and the crowdworker communities coexist to form a broader workplace.

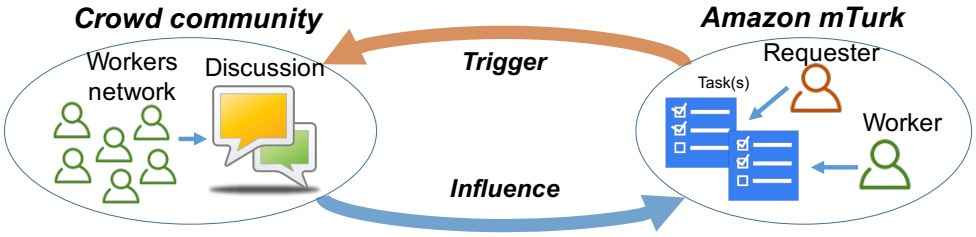


Figure 4.1: Bird's eye view of the socio-technical system that is built around Amazon Mechanical Turk.

Despite being "invisible" to microwork platforms, crowdworker communities share a great amount of relevant information about HITs, requesters, and their own work [200]. Yin et al. [238] highlight the presence of rich network topology around crowdworkers, which is built around online fora. Authors enabled their analysis by injecting a HIT in mTurk, to provide crowdworkers incentive to self-report their connections. In [148] an ethnomethodological study was conducted on the crowdworker community active on `TurkerNation`. Researchers observed crowdworker discussions on the forum for a period of seven months, analysing crowdworkers' motivation and their attitude towards the different actors in the marke (especially requesters). Their work clearly shows that crowdworkers regard their activities on mTurk as paid work, and, often, as main source of income. Therefore, crowdworkers strive for efficiency in work execution and fairness, and transparency in the way the work is evaluated and rewarded.

These studies inspired and informed our work, but suffer from several limitations. **1)** They analyse a limited amount of fora (often a single one), thus focusing on a subset of crowdworker communities. **2)** They offer an analysis that is limited in time. **3)** They do not analyse the relationship with the microwork market status. As our results demonstrate, fora are differently popular, and the activities of their members encode different norms, preferences, and behaviours. In our work, we scale up the analysis to multiple fora, considering their whole history of existence. The resulting dataset combines information from the mTurk market, and it is unique both in scale and diversity. Finally, our quantitative analysis has a broader spectrum, and it includes the study of the relationships between activities in crowdworker fora and the properties, availability, and consumption rate of HIT groups in the mTurk market.

4.3 Dataset

We consider the 4 most popular general-purpose fora related to mTurk, namely `mTurkCrowd`, `mTurkForum`, `TurkerNation` and `mTurkGrind`. The study also includes `Turkopticon` and `Reddit HWTF`, two popular specialized fora online services for crowdworkers. `Turkopticon` [105] is a system designed to focus on the evaluation of *requesters* (and their HIT groups), as performed by mTurk crowdworkers, who are also allowed to comment on ratings. `Reddit HWTF` (HITsWorthTurkingFor) is a subreddit devoted to the advertisement of HIT groups that community members deem particularly worth of attention.

Given the domain-specific nature of these 6 fora, we assume all their members to be also workers in mTurk, including administrators and moderators. We are aware, from discussions with administrators of several fora, that some of the registered members are requesters in mTurk, or scientists interested in studying crowdworker activities. We believe their number to be limited, and their impact on the forum activity to be marginal for the purpose of this study. We cannot exclude that crowdworkers might participate in multiple fora. However, results from previous work show that workers are more likely to communicate with others from the same forum [238].

4.3.1 Dataset Creation

Fora. Our analysis focuses on content that is publicly visible on fora, or available to registered users. We retrieved the whole history of discussions and messages of `mTurkCrowd`, `mTurkForum`, `TurkerNation`, `mTurkGrind`, and

Forum	**Start**	**#Members**	**#Threads**	**#Messages**	**Msg./Mem.**	**Msg./Thr.**	*Purpose*
mTurkForum	07/12	4,926	1,889	1,427,856	289.9	755.9	General
mTurkGrind	10/13	3,217	943	948,775	294.9	1006.1	General
TurkerNation	11/14	572	563	173,942	304.1	309	General
mTurkCrowd	01/16	616	131	177,669	288.4	1356.3	General
Turkopticon	01/09	18,640	310,129	371,750	19.2	5.1	Evaluation of requesters
Reddit HWTF	03/16	930	3,490	17,843	19.9	1.2	HITs advertisement
Total		28,901	317,145	3,117,835	202.7	572.3	

Table 4.1: Descriptive statistics of the six targeted fora. Legends: **Start** – earliest crawled message.

Forum	**Linked HIT Groups**					**Linked Requesters**				
	#HM	AvgHMW	% HMF	#HITs	% MH	#RM	AvgRMW	% RMF	#REQ	% MR
mTurkForum	233,294	47.36	16.30%	104,893	15.81%	217,489	44.15	15.19%	22,737	43.08%
mTurkGrind	229,504	71.34	24.08%	100,310	23.07%	233,063	72.45	24.46%	22,078	60.44%
TurkerNation	73,637	128.74	41.87%	41,659	5.44%	73,767	128.96	41.95%	11,610	45.69%
mTurkCrowd	40,771	66.19	21.88%	19,278	11.76%	41,522	67.41	22.29%	6,896	85.66%
Turkopticon	NA	NA	NA	NA	NA	371,797	19.95	94.50%	45,701	54.10%
Reddit HWTF	1,937	2.08	10.01%	1,649	2.58%	635	0.68	3.28%	415	9.21%
Overall	579,143			184,390		938,273			50,912	

Table 4.2: Statistics of links to mTurk HIT groups and requesters discovered in fora. Legend: **#HM** – number of messages with Links to HIT groups; **AvgHMW** – average number of HM per user in forum; **% HMF** – percentage of HM in forum messages; **#HITs** – number of unique HIT groups mentioned in forum messages; **%MH** – percentage HIT groups in the market mentioned in messages; **#RM** – number of messages with Links to Requesters; **AvgRMW** – average number of RM per user in forum; **% RMF** – percentage of RM in Forum messages; **#REQ** – number of unique Requesters mentioned in forum messages; **%MR** percentage of requesters in the market mentioned in messages. *Ratio of mentioned HIT groups and Requesters are calculated within the timespan of existence of each forum.*

`Turkopticon` (until May 20th 2016) using custom Web crawlers. The **Start** column in Table 9.1 contains, for each forum, the creation date of the earliest crawled message. `Turkopticon` is the oldest forum in the pool (January, 2009)[2]. Reddit content was retrieved using the official reddit API, which, unfortunately, sets limitations in the amount (and age) of accessible content. Therefore, our collection of `Reddit` HWTF posts is limited to the period Mar. 27, 2016 to May 20, 2016.

Table 9.1 reports descriptive statistics of the resulting dataset. For each thread and message, we retrieved title, content, time-stamp, and creator. We collected more than 3.1M messages, produced by 28.9K members that published at least one message. Non-human members (bots) and their messages have been identified and removed from the dataset.

mTurk Market. We operated on a dataset that spans 6 years of Amazon mTurk activities. We started with the dataset studied in [63], which contains more than $2.56M$ distinct HIT groups, and $130M$ HITs produced from 2009 to 2014. To analyse the activities of more recent fora, we enriched the dataset with 46K HIT groups and 1.9M HITs collected between Apr. 11 and May 20 2016. All HIT groups in the dataset are described by metadata, including their size at publication, title, description, reward, and allotted time. To study the evolution of HIT groups consumption over time, we adopted the notion of HIT group throughput proposed in [63], that is, the number of HITs in the group completed in a given time interval. Throughput information is obtained by periodically crawling (every 5 minutes) the mTurk system, to retrieve, for each active HIT group, the amount of available HITs.

4.3.2 Message Categorization

For a deeper understanding of workers' communications, we categorize messages in the dataset according to the type of discussion they include. Given the size of the dataset, we resorted to supervised learning for automatic classification. A manual annotation process was instrumented to create a training set of suitable size.

Annotation of Training Dataset To minimize sampling bias, we randomly selected 10% of all threads from each forum, except `Turkopticon`. From each selected thread, we picked a random sample of at most 50 messages. In `Turkopticon`, given the amount (and topical homogeneity), we sampled

[2]It is important to mention that `TurkerNation` is the new instantiation of an older forum (http://turkers.proboards.com) that migrated technological platform in 2014. We were not able to retrieve earlier data.

500 threads. The resulting 13,017 messages were manually inspected to label messages. We employed card sorting, a technique widely used in the design of information architecture to create mental models and derive taxonomies from input data [203]. From recent work on crowdworker communities we elicited a number of message types (e.g. "problems, suggestions, tips" and "community communication and interests" from [148]). Then, using *open* card sort, we synthesized and defined six types of messages, described below:

- **Ask or Answer**: messages that include questions asked by a crowdworker, or answers to previous questions. This category included messages inquiring for general purpose issues with mTurk or forums (e.g. how to obtain qualifications), or seeking for explanations about tasks. *Example*: "Anyone able to withdraw?"
- **Comment**: messages that include general comments about a HIT group, such as its availability, requirements, or presence of bugs (e.g. lack of completion code). *Example*: "Can't be on mobile"
- **Experience**: messages that report the experience of a crowdworker in the execution of a HIT, e.g. the amount of time spent on a task, or the amount of rewarded bonus. *Example*: "Projected Earnings for Today $70.00"
- **Judgement**: messages where crowdworkers explicitly express compliment or criticisms about HIT groups or requesters. *Example*: "$0.60 cent one is good, 0.36 hit sucks"
- **Rating**: messages that include a reference to `Turkopticon` rating, or rating in other fora. Rating messages often serve as recommendation from crowdworkers to the community, as only HIT groups worthy of discussion are mentioned. *Example*: "This requester has actually joined Opticon just to flag negative reviews and accuse them of blackmail."
- **Social**: messages where crowdworkers address the community with general-purpose social topics, such as greetings and jokes. *Example*: "Turtles for days Happy new year!"

We then applied closed card sort to categorize all messages in the training set. We created, in a digital form, a card for each message. By means of an online collaboration tool, other researchers (including the second and last author) were involved in assigning cards to message types. To reduce bias and strengthen the validity of results, all researchers reviewed and agreed upon the categorization of messages. Messages could belong to multiple types. For instance, it is common for workers to rate a task while sharing information about their experience, or expressing a judgment about the task.

The resulting training dataset shows a clear skewness in the frequency of message types. 54.55% of messages were classified as *Social*, 28.45% as *Rating*, 20.53% as *Experience*, 12.80% as *Judgement*, 3.32% as *Comment*, and 11.24% as *Ask or Answer*.

Automatic Classification. We fed a multi-label Random Forest classifier with textual features of the annotated messages (bag of words, TF-IDF weighted) and trained to predict the message's type. To account for the relative sparsity of some message types in the training dataset, we assess the performance of the classifier both in terms of accuracy and F-score in a 5-fold cross-validation setting. The classification performance is reported in Table 4.3.

Type	Accuracy	F-Score	Type	Accuracy	F-Score
Ask or Answer	0.86	0.27	Comment	0.98	0.60
Experience	0.80	0.46	Judgement	0.86	0.46
Rating	0.93	0.85	Social	0.75	0.74

Table 4.3: Performance of message type classification.

Rating, *Social*, and *Comment* messages are those identified more accurately by the classifier. The classification of *Experience* and *Judgement* messages is also accurate and with acceptable F-score. *Ask or Answer* messages are the most difficult to classify, with high accuracy but low F-Score. We therefore exclude this category of messages from subsequent analysis, and leave the improvement of classification performance to future work.

Analysis of Message Categories Distribution. 54.95% of messages in the dataset were classified as *Social*; 25.35% were classified as *Rating*, 18.35% as *Experience*, 8.67% as *Judgement*, and 4.90% as *Comment*. The distribution of message types is consistent with the the result of manual annotation. The distribution also reveals that the amount of crowdworkers' social-related activities is comparable to their work-related activities (Rating, Experience, etc.). This result quantitatively supports the outcome of previous work [126], and highlights the dual nature of online crowdworker communities, where both social and technical needs are addressed. Notably, the *Judgement* message type has a relatively low frequency, compared to *Rating*. This suggests a preference for standard ways (i.e. `Turkopticon` ratings) to express opinions about requesters and HIT groups. An analysis of the linguistic properties (e.g. sentiment) of such judgment is an interesting topic for future work.

Figure 4.2 shows the (log scale) distribution of message categories across fora. It is possible to observe the specular frequency of *Comment* and *Judge-*

		Group Size	Reward (cents)	Time (minutes)	Requirement	Task Type %SU	%CC	%CA	%IA	%VV	%IF	%OT
mTurkForum	**M**	450.31 ± 3895.49, 1	90.62 ± 275.54, 50	70.05 ± 153.93, 60	0.55 ± 0.50, 1	68.77	15.08	0.12	7.85	1.08	6.62	0.50
	UM	34.79 ± 419.63, 1	378.31 ± 792.26, 52	131.22 ± 206.12, 60	0.14 ± 0.35, 0	19.66	55.37	1.24	4.69	15.68	3.27	0.08
mTurkGrind	**M**	493.39 ± 3422.52, 1	66.65 ± 344.92, 20	72.95 ± 177.07, 45	0.55 ± 0.50, 1	47.54	18.48	0.51	17.97	4.31	10.37	0.82
	UM	30.43 ± 370.04, 1	381.19 ± 793.75, 55	131.64 ± 205.98, 60	0.14 ± 0.34, 0	19.78	55.61	1.24	4.42	15.72	3.16	0.07
TurkerNation	**M**	541.55 ± 3563.56, 1	91.88 ± 285.69, 50	77.26 ± 174.98, 60	0.61 ± 0.49, 1	63.87	15.14	0.11	11.27	1.22	7.96	0.44
	UM	30.14 ± 366.03, 1	380.09 ± 794.05, 52	131.44 ± 206.02, 60	0.13 ± 0.34, 0	19.46	55.62	1.25	4.59	15.77	3.22	0.08
mTurkCrowd	**M**	367.00 ± 2599.24, 1	74.95 ± 210.28, 40	91.97 ± 212.11, 60	0.59 ± 0.50, 1	62.98	17.09	0.79	9.6	1.31	7.76	0.47
	UM	25.60 ± 320.21, 1	388.21 ± 802.23, 56	132.10 ± 205.04, 60	0.12 ± 0.33, 0	18.38	56.55	1.24	4.51	16.13	3.11	0.07
Reddit HWTF	**M**	123.13 ± 645.40, 1	46.51 ± 65.62, 32	55.75 ± 83.50, 60	0.65 ± 0.48, 1	83.30	5.03	0.24	5.72	0.46	5.03	0.22
	UM	40.43 ± 643.03, 1	377.28 ± 791.02, 53	131.05 ± 206.31, 60	0.14 ± 0.35, 0	19.76	55.27	1.23	4.73	15.61	3.31	0.09

Table 4.4: Descriptive statistics – mean (μ) ± standard deviation (σ), and median (m) – of metadata, and task type distribution for *mentioned* (**M**) and *unmentioned* (**UM**) HIT groups . **Task Type**s: SU – Survey; CC – Content Creation; CA – Content Access; IA – Interpretation and Analysis; VV – Verification and Validation; IF – Information Finding; OT – Other types. Differences within fora are statistically significant (Mann-Whitney test, p-value $< .001$) for all the analysed properties.

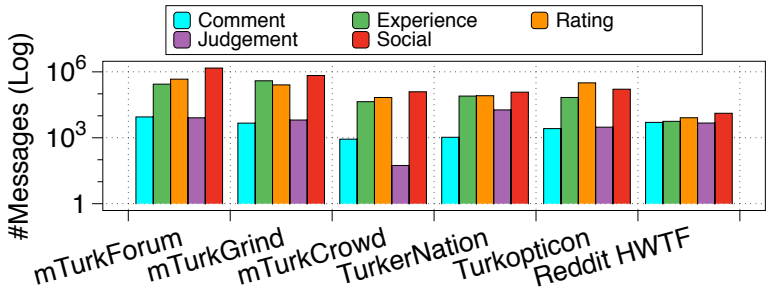


Figure 4.2: (Log scale) distribution of message types (in #Messages) in fora after the classification of all the messages.

ment messages in `mTurkCrowd` (the youngest forum) and `TurkerNation`, both fora with similar amount of members and conversation, but with different social norms [126]. Members of `Turkopticon` mainly post *Rating* messages, although the amount of *Social* messages is interestingly high, considering the specialized nature of the forum. Members of `Reddit HWTF` do not favour a single message type, thus giving equal importance to all aspects related both to work and social interactions. Overall, it would seem that each forum caters for different discussion needs, and focuses on different, specific topics. This finding will be taken into account in the following analysis.

4.3.3 Linkage to mTurk

To study the relationship between the activities in fora and the evolution of HIT groups availability and consumption over time in the mTurk market, HIT groups and requesters must be identified in the published messages. We focus on explicit *mentions*, i.e., unambiguous references to HIT groups and requesters. To this end, we instrumented a text analysis pipeline that parses the text in threads and messages to extract and process `http` links towards mTurk pages of HIT groups and requesters.[3] This technique allows us to achieve maximum linking precision.

Of all messages in the dataset, 22.84% link to at least one HIT group, while 33.62% link to a requester page in the mTurk market. Table 4.2 reports the distribution of links across fora. We retrieved a total of 184K distinct HIT groups (up to 20% of the total amount of groups available in the market

[3]Links reference HIT groups (i.e. https://www.mturk.com/mturk/preview?groupId=<tID>, where <tID> is the identifier of a HIT batch) or requesters (i.e. https://www.mturk.com/mturk/searchbar?selectedSearchType=hitgroups&requesterId=<rID>, where <rID> is the identifier of a requester).

during the considered fora lifetime) from 579K messages, and 51K distinct requesters (up to 85% of active requesters) from 938K messages.

4.3.4 Coverage of HIT Groups and Requesters

Table 4.2 summarizes the distribution of links to mTurk HIT groups (**%MH**) and requesters (**%MR**) across fora. The considered fora cover a partial, diverse, yet numerically significant share of the mTurk market.

It is possible to observe significant differences in terms of market coverage. For instance, the HIT groups mentioned in `TurkerNation` account for only 5% of the market. Fora like `mTurkForum` and `mTurkGrind` feature a better coverage (respectively 15% and 23%). Such differences can be partially explained by forum-specific "culture" and norms. For instance, the low recall of `Reddit HWTF` (less than 3% of the HITs in the market) can be explained by its mission statement[4], where members are asked to only report HIT groups with fair hourly retribution. Similar observations hold when considering the distribution of linked requesters. The high coverage obtained by `mTurkCrowd` (85.66%) could be explained by an interesting practice pursued by forum members: when a new requester posts work to mTurk, crowdworkers email the requester asking clarifying questions with the ultimate goal of determining if the new requester will be responsive to crowdworkers' concerns [126]. The low coverage (in terms of requesters, %MR) in `Turkopticon` could be explained by its age: as the participation in `Turkopticon` and the awareness of its utility grew over time, requesters operating in the early stage of the mTurk market were more likely to be "ignored" by `Turkopticon` members.

4.4 The Influence of the Market on Fora Discussions

In this section we address **RQ1**, and investigate how the content and the dynamics of the mTurk market influence discussions in crowdworker fora. As dimensions of analysis, we consider **1)** properties of published HIT groups; **2)** temporal variations in the mTurk market demand; and **3)** properties of HIT groups' requesters.

[4]https://www.reddit.com/r/HITsWorthTurkingFor/wiki/index

Forum	**F-Statistics**	p**-value**	**Opt. Lag** (Minutes)
`Social`	3.0664	0.0001	225
`Rating`	2.8737	0.0002	225
`Experience`	2.7180	0.0003	240
`Judgement`	1.5173	*0.0510*	360
`Comment`	3.6913	0.0006	135

Table 4.5: Synchronization between the HITs availability in the market and HITs mentions across different message types.

Forum	**F-Statistics**	p**-value**	**Opt. Lag** (Minuts)
`mTurkForum`	5.8429	0.0006	45
`mTurkGrind`	1.5599	*0.0663*	255
`TurkerNation`	1.8398	0.0166	270
`mTurkCrowd`	2.7972	0.0002	225
`Reddit HWTF`	2.3129	0.0080	225

Table 4.6: Synchronization between the HITs availability in the market and HITs mentions across different fora.

4.4.1 HIT Groups Properties

We analyse five properties of a HIT group: 1) *Group Size*, i.e. the amount of HITs available at publication time; 2) *Reward*, i.e. the amount of monetary compensation associated with a successful execution of a task; 3) *Time Allotted* for task execution, as specified by the requesters; 4) *Requirement*, a boolean variable[5] that encodes the specification of as approval rate threshold for the worker to be allowed to execute the HIT; and 5) *Task Type*, defined according to the taxonomy proposed in [70]. Table 4.4 reports descriptive statistics for HIT groups that are mentioned (respectively, unmentioned) by community members of different fora. The analysis provides a number of non-trivial insights, showing both the influence of the market on fora, and the heterogenous nature of fora communities.

1) The distribution of task type popularity in mentioned hits significantly differs from the distribution of task availability in the market. For instance, *Survey* is the most favoured task types in all fora, while previous work [63] reports that *Content Creation* is the most available task type. The result highlights a clear difference between work demand in the mTurk market and the preference of community crowdworkers. **2)** There are relevant differences in the popularity of task type across fora. For instance, IA and VV task

[5] A value of 1 encodes "Qualification or approval rate greater than x needed".

types are more popular in `mTurkGrind`, while `Reddit HWTF` emerges as the most polarized towards *Survey* tasks. The result suggests a forum-specific "interest profile" for task types. **3)** The properties of unmentioned tasks show no statistically significant difference across fora (Mann-Whitney test, p-value $> .001$). The result suggests the presence of a shared rejection of HIT groups having traits that are not considered worthy of discussion. The presence of a common rejection baseline gives more value to the differences that emerge when analysing mentioned HIT groups. **4)** HIT groups are more likely to be mentioned when having large size, shorter time allotted, requirements for execution, and lower reward. The result suggests that crowdworkers are more likely to discuss HIT groups if there is opportunity for large amounts of work to be performed, or if there are limitations in their ability to execute tasks. Issues about rewards are not deemed important. We argue these results to be a clear indication of the dominant role that the market has on the task selection strategy of communities, where the need for guaranteed income prevails over issues of fair payment – alas an accepted (yet unpleasant) norm. Attitude towards (un)fair payment seems to greatly vary across fora. For instance, tasks mentioned by members of `mTurkGrind` are up to 38% less rewarding than the tasks preferred by other communities.

Likewise, `Reddit HWFT` members are less concerted with group size and with reward, surprisingly betraying the original mission statement of the forum. **5)** Preferences for mentioned HIT groups often differ in a statistically significant fashion across fora. Differences in the distributions of `Group Size` properties are statistically significant across all fora (Mann-Whitney test, p-value $< .001$); differences of `Reward` values are significant between `mTurkGrind` and all fora (Mann-Whitney test, p-value $< .001$), `mTurkForum` and `Reddit HWFT` (Mann-Whitney test, p-value $< .001$), and between `mTurkCrowd` and `Reddit HWFT` (Mann-Whitney test, p-value $< .001$).

4.4.2 Task Availability in the Market

In this section we investigate the relationship between the dynamic properties of the mTurk market, and the discussion by crowdworker communities. We compare the temporal distribution of the amount of HIT groups *available* in the market, with the temporal distribution of the amount of HIT group mentions in fora.

Analysis of Aggregated Fora Activities. First we analyse the temporal distribution of mentions across all fora, to study the relationship between the availability of HIT groups in the mTurk market and the whole set of

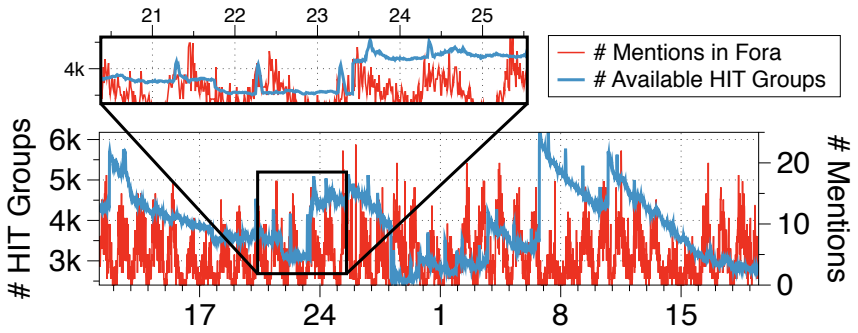


Figure 4.3: Time series of available #HIT groups in mTurk, and #Mention by crowdworker communities. Period: Apr.-May 2016.

online crowdworker communities. The analysis includes market data related to time intervals where all fora were active – i.e. three months, March 2016 – May 2016. Figure 4.3 shows an example of the two time series. As in previous work [63], we observe a weekly periodicity for the market work demand (HIT groups availability). A similar periodicity is observed in the temporal distribution of mentions, with higher volumes of messages posted during weekdays. We found evidence of daily periodicity in both HIT groups availability and number of mentions, with a peak in the early morning (PST time). Compared with the work demand curve, the temporal evolution of discussions (mentions in fora) shows smaller variations across days. This is an interesting finding, that suggests the presence of an *upper bound* in the amount of HIT groups that can be discussed (or that are worthy of discussion) that is only partially dependent on the current market demand.

Co-locating the two time series on the time axis, we observe that the peak time of the #Mentions distribution closely relates with the amount of HIT groups available in the market, with a delay in the range of 1 to 6 hours.

We thus hypothesize that there exists a quantifiable relationship between the dynamics of work demand in the market and work-related discussions in crowdworker communities. To test this hypothesis, we test for Granger Causality [81] between the two time series. Granger Causality is a technique for determining whether one time series is significant in forecasting another [64]; in other terms, Granger Causality measures a statistical dependence between the past of a process and the present of another. It is a statistical test widely used in fields such as econometrics, data mining [11] and machine learning [172]. In Granger Causality, a "lag" parameter captures the temporal delay between the two series for which better prediction is achieved. Optimal

lag is usually selected by searching for the one with the lowest AIC/BIC (Akaike or Bayesian information criteria [27]) within a predefined range.

Since the time series of available HIT groups shows large variations across days, we first de-trend its temporal distribution by applying Z-score normalization on a daily basis, such that similar variance can be obtained in different days. Z-score is a normalization technique that is often adopted in time series analysis [24] to interpret the magnitude of time series' fluctuations in terms of a common scale.

The analysis provides two relevant insights. **1)** The temporal distribution of the #mentioned HITs in the fora is significantly correlated with the de-trended distribution of number of available HIT groups (F-Statistics: 2.8681, p-value = .002).[6] The result confirms the presence of a *quantifiable* relationship between work demand in mTurk and work-related discussions. **2)** The highest correlation is achieved with a lag value of 4 hours. The result indicates that variations in market demand have a visible effect on the activity of crowd communities after **4 hours** (on average) .

Analysis of Message Categories. We then investigate the presence of temporal causality between market demand and fora messages that mention HIT groups with a specific message type. Results shown in Table 4.5 indicate that market demand is a significant predictor for the temporal distributions of all message types (with the exception of *Judgement*). *Comment* messages are the ones for which stronger prediction power (F-Statistics) and shorter delay (**2 hours** delay) can be observed. As *Comment* messages include information about task requirements and work-related issues, this result provides an indication of the minimum (yet averaged, across fora) "reaction time" that crowdworker communities can have to variations in market demand. *Social*, *Rating*, and *Experience* messages show an additional delay of 90-120 minutes. This result suggests that discussions about work execution temporally (but not quantitatively[7]) precede communication for other purposes.

Analysis per Forum. Finally, we address differences in temporal correlation across the considered fora. Results in Table 4.6 show that discussions in all fora (except `mTurkGrind`) are significantly correlated with market demand. The distribution of values in Table 4.6 highlights a correlation between the

[6] We stress how, in the context of our work, the Granger Causality test proves a temporal synchrony between the two series, *but does not fully prove causality.* This is obvious, as we could not control, in our collected data, for other temporal and contextual factors possibly influencing the HIT mentions in crowdworker communities.

[7] The type distribution of messages linked to HIT groups is heavily skewed: *Rating*: 97.14%; *Social*: 63.64%; *Experience*: 57.15%; *Judgement*: 45.63%; *Comment*: 3.34%.

amount of forum members and the temporal lag w.r.t. the market demand curve (One tailed Mann-Whitney U Test. p-value $< .01$). The task availability time series has stronger prediction power on `mTurkForum`, while the lower is with `TurkerNation`. The lack of significant synchronicity for `mTurkGrind` is of interest, given the age and popularity of the forum. We hypothesize that this result is due to difference in the distribution of preferred task types (CC, IA, and IF) HIT groups have an higher popularity than in other fora. Further investigations are left to future work.

		Communica.	Generosity	Fairness	Promptness
`mTurkForum`	**M**	3.35±1.41,3.50*	3.35±1.00,3.38*	4.42±0.87,4.85	4.40±0.80,4.71
	UM	3.19±1.50,3.20*	3.03±1.34,3.00*	4.24±1.20,5.00	4.21±1.14,4.81
`mTurkGrind`	**M**	3.37±1.37,3.50*	3.45±0.96,3.46*	4.45±0.80,4.85	4.41±0.76,4.69
	UM	3.19±1.51,3.20*	2.99±1.32,3.00*	4.23±1.20,5.00	4.22±1.12,4.80
`TurkerNation`	**M**	3.39±1.39,3.57*	3.44±0.96,3.46*	4.48±0.80,4.89	4.41±0.79,4.71
	UM	3.17±1.50,3.15*	3.01±1.31,3.00*	4.21±1.19,5.00	4.22±1.10,4.77
`mTurkCrowd`	**M**	3.40±1.40,3.62*	3.42±1.05,3.50*	4.47±0.87,5.00*	4.43±0.83,4.76*
	UM	3.12±1.49,3.00*	2.95±1.28,3.00*	4.18±1.18,4.86*	4.17±1.10,4.67*
`Reddit HWTF`	**M**	3.58±1.36,3.86*	3.81±0.80,3.86	4.64±0.64,4.91	4.60±0.59,4.81*
	UM	3.25±1.45,3.33*	3.16±1.20,3.17	4.32±1.05,4.95	4.29±0.99,4.75*
`Turkopticon`		3.27±1.45,3.40	3.21±1.17,3.22	4.32±1.05,4.93	4.30±0.98,4.75

Table 4.7: Descriptive statistics – mean (μ) ± standard deviation (σ), and median (m) – of reputation scores for *mentioned* (**M**) and *unmentioned* (**UM**) requesters. Properties that are significantly different within forum (Mann-Whitney test, p-value $< .001$) are market with *.

4.4.3 Requesters Properties

Finally, we investigate the relationship that exists between requesters in the mTurk market, and discussions in fora. We consider as dependent variables the reputation scores assigned to requesters on `Turkopticon`. Such scores include: 1) *Communicativity*; 2) *Generosity*; 3) *Fairness*; and 4) *Promptness*. Table 4.7 reports descriptive statistics for requesters that are mentioned (respectively, unmentioned) by crowdworkers of different fora. As reputation scores are obtained from `Turkopticon`, we included in the "unmentioned" group those requesters that are part of the `Turkopticon` database, but not mentioned by the analysed forum. Requesters that are more communicative and generous are consistently preferred in all fora. Notably, the difference in terms of *Communicativity* and *Generosity* between mentioned and unmentioned requesters is quantitatively similar across fora (respectively ~ 0.3 and

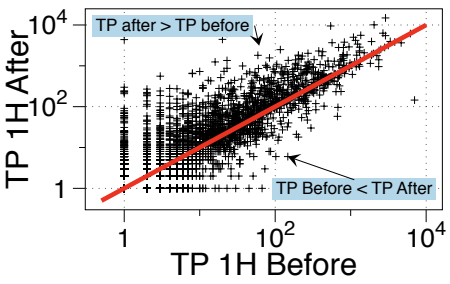
~ 0.4 on a 5 point scale). Crowdworkers in `mTurkCrowd` are the most selective in terms of *Fairness* and *Promptness*. This result could be explained by the practice of inquiring novel requesters discussed in Section 4.3.4. *Generosity* and *Fairness* values are significantly higher for requesters mentioned in `Reddit HWTF`; this could also be explain by the forum's mission statement. It is also interesting to observe the high *Fairness* values in `Turkopticon`, and for both mentioned and unmentioned workers. This result strides with the findings in Section 4.4.1, where we observed workers favouring tasks with lower reward but comparable execution time. Considering that previous results highlight the importance of task complexity [232], we hypothesize that the value of *Fairness* may relate to intrinsic properties of tasks, that are not easily observable from HIT groups metadata. The investigation of this hypothesis is left to future work.

4.5 The Impact of Community Activities on Tasks Consumption

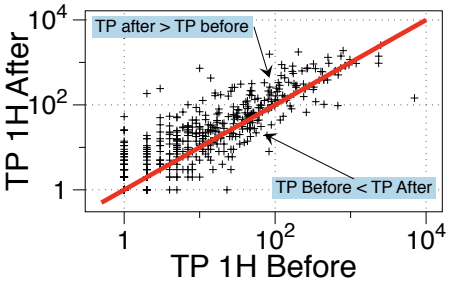
In this section we address **RQ2**, and investigate the presence and effect of a quantifiable relationship between discussions about HIT groups in fora, and the *consumption speed* (throughput)[8] of such groups in mTurk. First, we investigate the presence of significant differences between the average throughput of HIT groups that were mentioned in fora, and the average throughput of HIT groups that were not mentioned. Then, we seek stronger evidence of temporal correlation by using the same time series analysis technique introduced in the previous section. We consider the progressions of throughput at individual HIT group level, and compare them with the respective temporal distribution of their *mentions* in fora.

Analysis of Throughput Differences for Mentioned HIT Groups. To perform the analysis, we need to identify HIT groups featuring enough data points describing both the fora *mention* and the task *consumption speed* time series. The dataset developed in [63] contains consumption data for 149K HIT groups; the addition of data from the 46K HIT groups collected between May and March 2016 yields a total of $195,332$ HIT candidate groups. From this set, $26,204$ groups are linked to *mentions* in fora. The task type distribution of these $26,204$ HIT groups is however different from the set of HIT groups having at least one mention (184K, Section 4.3.3). To improve

[8] The amount of HITs in a group that get completed between two successive observations (typically, every 1 hour).



(a) Mentioned HIT Groups



(b) Groups with Granger Causality

Figure 4.4: Hourly throughput (TP) HIT groups, 1 hour before and 1 hour after the first mention in fora.

the generalizability of results, we applied stratified random sampling (with strata corresponding to task types), to obtain an analysis dataset having comparable distribution. The result is a set of 19,122 HIT groups.[9]

We found a statistically significant difference (Mann-Whitney test, p-value $< .001$) between the average hourly throughput of mentioned HIT groups ($\mu = 27.35$, $\sigma = 286.14$, $m = 0.09$) and the average hourly throughput of unmentioned groups ($\mu = 19.53$, $\sigma = 137.95$, $m = 1$). The result suggests the presence of an acceleration effect due, at least in part, to mentions in fora. To better characterize this acceleration effect, we compare the HIT group consumption speed *one hour before* and *one hour after* their earliest mention in fora. Figure 4.4(a) shows that the consumption of the majority of tasks is boosted after worker discussions: on average, a 59.26% increase.

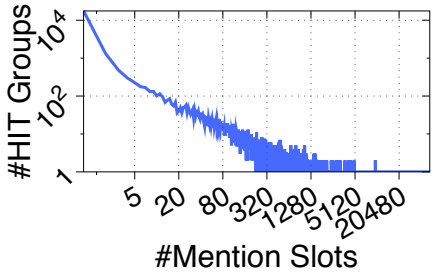


Figure 4.5: Distribution of #overlapping mention slots across HIT groups.

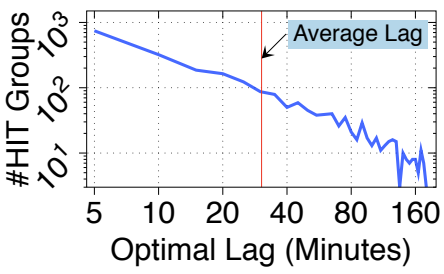


Figure 4.6: Distribution of optimal lags in Granger-Causal HIT groups.

[9]Task types distribution: 59% SU, 16% CC, 0.6% CA, 11% IA, 2% VV, 8% IF, 0.3% Other.

Analysis of Temporal Correlation. We apply Granger Causality analysis to investigate the presence of temporal relation between HIT group mentions and throughput variations. We discretize the *mention* time series into 5 minutes slots, to align it with the sampling rate of HIT groups consumption. Figure 4.5 shows the heavy tailed distribution of the number of time slots that feature an overlap between the *consumption* and *mention* time series of a HIT group.

We consider tasks with ≥ 5 overlapping slots. The resulting $4,539$ HIT groups feature an average hourly throughput of 22.10 ($\sigma = 269.80$, $m = 0.06$) and a comparable task type distribution (54.2% SU, 19.2% CC, 0.5% CA, 15% IA, 2.1% VV, 8.6% IF).

$1,541$ HIT groups (33.95% of the considered set) show significant Granger Causality (p-value $< .05$) between the temporal progression of their mentions in fora, and their consumption. When compared to the overall population of HIT groups mentioned in fora, these $1,5K$ groups have lower averaged hourly throughput ($\mu = 12.6$, $\sigma = 115.23$, $m = 0.10$), a slightly different distribution in terms of task types (51.2% SU, 20.7% CC, 0.4% CA, 16.1% IA, 2.4% VV, 8.9% IF), lower reward ($\mu = 44.65$, $\sigma = 130.50$, $m = 15$), higher allotted time ($\mu = 225.07$, $\sigma = 4591.08$, $m = 45$), and comparable group size.

The strongest causality is found when the average lag is set to 30.40 minutes ($\sigma = 36.62$ minutes). Figure 4.6 shows the log-log distribution of the optimal lag across HIT groups: the majority of groups achieve higher causality for lags *lower than **15** minutes*. The result shows that crowdworker discussions can have a quick effect on the market. In the attempt of providing a quantification of such effect, we compare the HIT groups consumption *one hour before* and *one hour after* their earliest mention in fora. Figure 4.4(b) shows that the consumption of the majority of tasks is boosted, on average, by **3.4** factor (340%) after crowdworker discussions.

4.6 Discussion

This section discusses the insights reported in Sections 4.4 and Section 4.5, in the light of the research questions defined in the introduction. Then, we discuss possible threats to validity.

RQ1. Results from Section 4.4 provide novel insights into the preferences of crowdworkers active in online communities. We found an interesting discrepancy between market demand (in terms of task types) and the mentions of HIT groups by crowdworkers in fora. *Survey* tasks are the most mentioned,

and we observed forum-specific preferences for other task types. Crowdworkers are more likely to discuss HIT groups if there is opportunity for large amounts of work to be performed, regardless of unfair hourly reward. Using time series analysis techniques, we found significant synchronicity between the temporal evolution of available HIT groups in mTurk, and mentions in fora. Peaks of market demand correspond to peaks in crowdworkers' discussion activity, with an average delay of 4 hours. There are significant differences in terms of prediction strength and average lag across fora and discussion categories. `mTurkForum` emerges as the forum where activities can be predicted by market variation with minimum time delay (45 minutes); *Comment* messages (i.e. messages with comments about HIT groups) feature the smaller time lag. In terms of requesters' properties, we found *Communicativity* and *Generosity* to be the only properties commonly valued across fora. Crowdworkers in `mTurkCrowd` also favour *Fairness* and *Promptness*.

RQ2. Results from Section 4.5 provide a quantification of the effect that activities in fora can have in terms of HIT groups consumption. We measured a statistically significant positive ($\mu = 42\%$) difference in the throughout of HIT groups mentioned in fora, and an average 59.26% increase of throughput in the first hour after the first mention in fora. Time series analysis revealed that, for HIT groups featuring temporal synchronicity with mentions in fora, the average lag is of 30 minutes, and a 340% average throughput increase in the first hour after the first mention in fora. It is worth noting that these HITs are characterized by lower reward and higher allotted time: as such, they may not be appealing for workers in the market at a first sight. By being referenced in fora, the popularity of these HITs greatly increases, as our analysis proves. This also suggests that crowd fora, as a whole, can count on more resources to scout and select HITs worth completing. Further work is needed to identify the characteristics that make those HITs worth recommending.

Threats to Validity Results could suffer from several types of crowdworker selection bias. Members of the selected fora might not be representative of the wider population of mTurk crowdworkers. This risk is mitigated by the popularity of fora among crowdworkers. Previous work shows that a relevant amount of workers (up to 60% in sampled population) are active in at least one fora [238]. However, we consider three additional biases due to: 1) the omission from the study of other mTurk fora;[10] 2) the homogeneity of crowdworkers' country of origin;[11] and 3) the overrepresentation of crowdworkers

[10] Authors are aware of less popular fora (e.g. "CloudMeBaby"), but decided to scope the analysis to the most popular ones.

[11] Crowdworkers in fora are mainly from the US [148].

sharing social and economic needs (e.g. the need for a guaranteed income) that are not relevant for other crowdworkers. While we can't exclude the presence of these biases, we must acknowledge the importance of the investigated communities, as they represent a considerable share of the mTurk workforce.

The validity of our results could be also threatened by the task and requester linkage procedure (Section 4.3.3). To maximize precision, we only rely on explicit links in messages, thus failing to consider indirect references (e.g. members referring to requesters only by name). During the training set annotation activity (Section 4.3.1) more than 13K messages were analysed, but we found only a minority of messages (approximately 5%) referring to HIT groups or requesters but not including a link. We therefore believe this limitation to have negligible impact on the validity of our results.

4.7 Conclusion

Crowdworker fora are a relevant part of the microwork ecosystem. In this work, we hypothesized that the activities in mTurk fora can influence – and can be influenced by – properties of the mTurk market and its actors. Based on a rich dataset linking 3.1M messages in online fora with mTurk 2.6M HITs groups, we found quantitative evidence of relevant relationships in both directions. The insights contained in this chapter are meaningful for a variety of aspect related to microtask crowdwork, e.g. the design of tasks, incentive and task allocation schemes, and novel microwork systems.

Part II

Task Modeling

This part focuses on task modeling. The concept of *task* is central to knowledge creation: requesters describe their demands for knowledge in the form of tasks, which is then executed by crowds for knowledge creation. Modeling task properties is thus the key for crowd knowledge creation acceleration. In this part, we investigate three task properties, namely, the quality of task formulations, the complexity of tasks, and the clarity of tasks.

Poorly formulated tasks are less likely to solicit high-quality knowledge from contributors, thus hindering the overall knowledge creation process. Given the available signal of the quality of task formulations (i.e., question edits) in on-line knowledge crowdsourcing systems, chapter 5 begins our study on task formulation in these systems. To obtain a more direct view of important task properties that can significantly affect knowledge creation results, we then study task properties from the perspective of crowd perception. For this purpose, we switch the object of our study to human computation systems, which enable us to recruit crowds for assessing the considered task properties from their perspective. We focus on perceived task complexity in chapter 6 and task clarity in chapter 7, as both are found to be highly influential on the speed and quality of knowledge creation.

For each of the three aforementioned task properties, we aim at *1)* understanding its effect on knowledge creation; *2)* identifying low-level task features (e.g. words, for task description) that relate with the magnitude of the property; altogether, our work *3)* provides novel methods for measuring the property, and guidelines for better task design, which could ultimately lead to knowledge creation acceleration.

Chapter 5. Motivated by the large amount of task reformulations (i.e., edits) in on-line knowledge crowdsourcing systems, we start our study of task modeling by addressing the quality of task formulations. Through a qualitative study on the traces of task edits in StackOverflow, we find that the need for edits is highly indicative of the quality of task formulations. To further our understanding, we categorize task formulations of poor quality according to types of editing actions, such as adding task details, providing examples, clarifying the context, etc. Based on such categorization, we then propose a two-step approach for automatically detecting poorly formulated tasks, and afterwards, suggesting the most likely editing actions to be performed. By extensive validation on the StackOverflow dataset, we demonstrate the effectiveness of our approach in providing accurate reformulation suggestions.

Chapter 6. To understand how complexity is perceived and distributed over crowdsourcing tasks, we instrumented an experiment where we asked crowds to evaluate the complexity of 61 real-world re-instantiated human

computation tasks. We show that task complexity, while being subjective, is coherently perceived across workers; on the other hand, it is significantly influenced by task type. Next, we develop a high-dimensional regression model, to 1) assess the influence of three classes of structural features (metadata, content, and visual) on task complexity, and, ultimately, 2) to measure task complexity. Results show that both the appearance and the language used in task description can accurately predict task complexity. Finally, to demonstrate the utility of complexity as a task modeling property, we apply the same feature set to predict task performance, based on a set of 5 years-worth tasks in Amazon mTurk. Results show that features related to task complexity can improve the quality of task performance prediction.

Chapter 7. By surveying workers in the CrowdFlower platform, we unveiled the concerns of crowds in confronting unclear tasks, thus motivating the need for mechanisms that can predict and measure task clarity. We then propose a novel model for investigating task clarity based on two different constructs, namely, the *goal* and *role* clarity. Similarly to our study on task complexity, we conduct an experiment to acquire labels for task clarity from crowds, and show that task clarity is coherently perceived by crowds, and is affected by the type of the task. We then propose a set of features to capture task clarity, and show that these features can be used to accurately predict task clarity. Finally, we perform a long-term analysis of the evolution of task clarity on Amazon mTurk, and show that clarity is not a macro-property of the marketplace, but rather a local property of individual tasks.

In summary, this part contributes new understanding on important task properties that can influence the quality and speed of knowledge creation, and novel methods to automatically measure these properties, thus to assist task requesters for better designing tasks, for accelerated knowledge creation.

Chapter 5

Asking the Right Question in Community Q&A Systems

This chapter studies task formulation in on-line knowledge crowdsourcing systems. Specifically, we study question formulation in community question-answering (CQA) systems, by analyzing the traces of question edits to understand the elements of a well-formulated question. We then introduce a two-step approach to automatically detect poorly formulated questions, and to suggest necessary edit actions to improve the quality of question formulation. We further investigate the topical and temporal influence, and the effect of user's knowledge, familiarity with the platform on the quality of question formulations.

This chapter is published as "Asking the Right Question in Collaborative Q&A Systems" [228], by J. Yang, C. Hauff, A. Bozzon, and G.-J. Houben in Proceedings of the 25th ACM conference on Hypertext and Social Media, pages 179-189. ACM, 2014.

5.1 Introduction

Community question-answering (CQA) systems are highly popular Web portals where everyone can ask questions, and (self-appointed) experts jointly contribute to the creation of evolving, crowdsourced, and peer-assessed knowledge bases [26][169], often in a reliable, quick and detailed fashion. Examples of such portals are Yahoo! Answers[2] (for all kinds of questions) and StackExchange[3], which consists of a number of sub portals, each dedicated to a particular topic, such as travelling, mathematics or programming.

In CQA systems users (*askers*) post questions, and rely on other community members to provide a suitable solution to their information need. Potential *answerers* (users that answer questions) look through the list of existing questions, typically ordered by recency, and decide whether or not to contribute to ongoing discussions. Such decisions are influenced by a multitude of factors, including time constraints, quality and difficulty of the question, and the knowledge of the answerer. Users can often also *comment* or *vote* on existing questions and answers. Commonly, when satisfied, an *asker* can mark an answer as *accepted*, thus declaring her need satisfied. Incentives to answer are often based on gamification features of a platform, such as reputation points [10].

Although the median time until a first answer is posted in response to a question can be in the order of a few minutes (as shown for instance for StackOverflow [146]), more and more questions [12] remain ignored or without an accepted answer. Questions are unanswered when their meaning is not clear to the community members, or when it is not possible, given the available information, to understand the nature of the problem (e.g. the source code that produces a compiling error is missing). A good question should have enough details (but not too much), enough depth (without drifting from the core subject), examples (if applicable) as well as avenues already investigated by the asker [163]. Well-formed questions attract more high-quality answers than poorly formed questions, as subject experts are more likely to help users that already put some effort into finding an answer themselves [12, 146, 214].

We focus on StackOverflow[4], a CQA platform covering a large variety of topics related to the software development domain. Introduced in 2008, StackOverflow features more than 5 million questions, and 10 million an-

[2] http://answers.yahoo.com/
[3] http://stackexchange.com/
[4] http://stackoverflow.com/

swers provided by more than 2 million users[5]. To manage and increase the likelihood of good and useful answers, users are provided with editing functionality, which allows the improvement of questions based on the feedback from other community members. Edits usually happen in response to comments or answers, a process which might require several interactions (asker waits for comments or answers, adapts the question, waits again, etc.) and, ultimately, might cause the question to sink in the list of open issues.

Our work contributes a novel approach to improve the question formulation process. We envision a system that upon question submission, provides askers with feedback about the aspects of the question they need to change (improve) in order to phrase their needs in the *right* way. This in turn is more likely to attract the *right* answerers.

Here, we perform a first study to investigate the feasibility of this idea. In particular, we propose and evaluate the following two-step approach:

1. Determine whether the question is of high quality or whether it requires an *edit* (**Question Editing Prediction**).
2. When an edit is required, identify which aspect(s) of the question need(s) to be improved to turn it into a high quality question (**Edit Type Prediction**).

In the process, we address the following research questions:

- **RQ1**: To what extent are traces of question edits (and the lack of edits) indicative of well or poorly formed questions?
- **RQ2**: Given sets of properly/poorly formed questions, is it possible to automatically detect which category the question belongs to?
- **RQ3**: Is it possible to predict the type of action required to make a question "better", i.e. improve its quality?

Our results show that:

1. The need for edits is indeed indicative of a question's quality.
2. The need for a question to be edited can be predicted with high accuracy.

[5]These numbers are based on the StackOverflow data released in September 2013.

3. The identification of the type of required edit is much more difficult to predict: we classified edit types in three categories, and found that only one of them can be accurately predicted.

In the remainder of this chapter we first briefly cover related work in Section 5.2. Then, in Section 5.3 we present our methodology and developed hypotheses. The experimental setup and the experiments are presented in Sections 5.4 and 5.5 respectively. Finally, we discuss our findings and present future work in Section 5.6.

5.2 Related Work

Community question-answering systems have been emerging as important collective intelligence platforms. Domain specific CQA platforms such as StackOverflow are transforming the way people share experience, create knowledge and, ultimately, contribute to the evolution of a given field [217, 234].

Several works focused on the issue of question and answers quality in CQA systems, providing a solid scientific support to the premises of our work. Burns and Kotval [35] describe thirteen dimensions that can be used to distinguish questions, including answer factuality, complexity, and depth of answer needed. Dearman and Truong [57] surveyed 135 active members of the Yahoo! Answers platform, identifying the composition of the question as one of the main factors leading to its consideration by the community. Harper et al. [90] investigated predictors of answer quality in several CQA sites, identifying as relevant dimensions the question topic, type (e.g. factual, opinion), and prior effort (i.e. the requester clearly indicated a previous attempt to solve the problem). On a higher abstraction level, an investigation into StackOverflow identified four main types of questions [163]: *Need-To-Known*, *Debug/-Corrective*, *How-To-Do-It*, and *Seeking-Different-Solution*. Recent work has also considered the evolution of user behaviour over time: Ahn et al. [5] studied whether users learn to be better question askers over time, by correlating past actions (e.g. receiving upvotes or comments, accepting answers, etc.) with the quality of the subsequent ones. Past work has also investigated the nature of unanswered questions on StackOverflow [12, 146, 214] - two of the main reasons behind a question remaining unanswered are the lack of clarity and the lack of required information (source code, etc.).

Previous work has also focused on a variety of prediction tasks, including question difficulty prediction [89], question longevity, user expertise estimation and question recommendation. Anderson et al. [9] studied the

factors that contribute to the long-lasting value of questions in StackOverflow. Liu et al. [138] proposed a competition-based model for estimating question difficulty by leveraging pairwise comparisons between questions and users. Another area related to our work is the estimation of user expertise in CQA systems. In [240] it was found that the expertise networks in CQA systems possess different characteristics from traditional social networks, and based on this finding an expertise metric was proposed. Similar aspects were also studied in [112, 169]. Relevant examples of contributions addressing the problem of routing questions to the right answerer can be found in [139, 142] and [245].

To the best of our knowledge, no previous work has targeted the problem of question editing in CQA systems. Iba et al. [103] analysed editing patterns of Wikipedia contributors using dynamic social network analysis; although several observations are related to our setting, the nature and purpose of wikis is different from the one of CQAs. The type and nature of collaborative acts was studied in [212] on the specific example of users proposing novel mathematical problems, or contributing to their solutions. While providing important insights, [212] focused on a qualitative assessment of the collaboration problem. The application of those insights, e.g. by means of automatic analysis methods, was not investigated.

5.3 Methodology

This section describes our experimental methodology. We first discuss and present the types of question edits typically encountered on StackOverflow. Publicly available data dumps[6] contain the entire history of all questions posted to StackOverflow. Every revision of a question includes information about the editor (the asker or another user) and the time of the edit. We considered only questions whose question body was edited, thus ignoring changes in the title or in the tags.

Then, we discuss how we approached the *edit prediction task* as well as the *edit type prediction task* (Section 5.3.2). Finally, Section 5.3.3 presents a number of hypotheses, derived from our research questions of Section 5.1.

[6]https://archive.org/details/stackexchange

5.3.1 Common Question Edits

We first need to define when we consider a question to be of high and of low quality respectively.

A question is of high quality and thus **well formed** if:

1. it has not been edited in the past; and,
2. it has received at least two answers (the median number of answers for questions on StackOverflow).

Previous work [167] relies on the number of positive preferences (upvotes) as question quality indicator. Due to the significant correlation between upvotes and number of answers[7] we settled on the number of answers as indicator.

In contrast, we hypothesize that a question might be initially of **poor quality** if it does not receive an answer within 12 minutes after its publication (the median answer time on StackOverflow), or if it is edited one or more times before it receives the first answer.

However, not all edits are equal: a question may be edited by the asker herself or by a different StackOverflow user[8]; an edit can lead to a major change in semantics or be simply a correction of a spelling error or a reformatting of the question.

In order to gain qualitative insights, we first conducted a small-scale study aimed at eliciting the most important edit categories on StackOverflow. We define as *important* the first edit (in the sequence of edits) that is temporally followed by one or more answers.

We randomly selected 600 (*question,important edit*) pairs, and had three trusted annotators describing the nature of the observed changes. We found that most of our edits fall into one (or more) of the following eight categories:

- **Source code refinement**: the provided source code is modified; additions are more frequent than removal or truncation.

[7] In our dataset with 5M questions, we observed a linear correlation coefficient of 0.25, p-value$<$0.001.

[8] StackOverflow users are allowed to edit other users' questions after they reach a particular reputation level.

- **Context**: the asker provides additional context and clarifies what she wants to do/achieve, as well as information about the "bigger picture" of this question.
- **HW/SW details**: inclusion of additional details about the hardware and/or software used (software version, processor specification, etc.).
- **Example**: the asker provides examples of inputs, or describes the expected results.
- **Problem statement**: the asker clarifies the technical nature of the problem by posting an error message, stack traces or log messages.
- **Attempt**: the asker details the attempts she already made in order to solve the problem, either before posing the question or in response to comments or posted answers.
- **Solution**: the asker adds/comments on the solution found for the question. The StackOverflow community explicitly encourages contributions where the user asking the question also provides the final answer. Some askers append their solutions, others create an answer in the discussion.
- **Formatting**: the asker fixes small issues including spelling errors and code formatting.

Table 5.1 provides an example of each edit type found in our data set (described in detail in Section 5.4), apart from the *formatting* category. This initial study shows that the most important edit types are related to question clarification as well as to the description of attempts made to solve the problem - including the working solution. We therefore decided to not further consider the *formatting* category.

5.3.2 Predicting Edits and Edit Types

Extracting Useful Question Edits The purpose of this step is to create the training and test data sets for our experiments. Our goal is to create a data set characterized by the presence of two distinct classes of questions, which will be used to train a classifier able to properly identify *edited questions* from *non-edited questions*.

Edited questions were selected as follows. Let there be n edits of question Q_i expressed as revisions $R^{i_1}_{t_{a_1}}, ..., R^{i_n}_{t_{a_n}}$. Here, Q_i can also be considered as

Edit Category	Post ID	Added Text (Excerpt)
Attempt (1st edit)	9943644	Update 1: I've tested the application with NHProf without much added value: NHProf shows that the executed SQL is ...
HW/SW details (1st edit)	7473762	I'm running OS 10.6.8

Source code refinement (1st edit)	13318757	Here is the code:
		`import android.content.Context;`
		`import android.graphics.Matrix;`
		`...`

Problem statement (1st edit)	7500461	The Error:
		`Exception in thread "AWT-EventQueue-0" com.google.gson.JsonParseException:`
		`The JsonDeserializer com.google.gson.DefaultTypeAdapters$Collection`
		`TypeAdapter@4e76fba0 failed to deserialize json object ....`

Example (1st edit)	11875006	I have a list of numbers like this in PHP array, and I just want to make this list a little bit smaller. `2000: 3 6 7 11 15 17 25 36 42 43 45 ...`
Context (1st edit)	13923053	EDIT: I have 'jquery-1.8.3.min.js' included first, then I have the line $.noConflict();. Then I have includes for external files using the prototype framework. then I include my user defined function and finally call it. But, I figured ...

Solution (2nd edit)	9215463	**EDIT 2: **Okay that's done the trick. Using @Dervall 's advice I replaced the MessageBox line with a hidden window like this:
		`MSG msg;`
		`HWND hwnd;`
		`WNDCLASSEX wcx;`

Table 5.1: Each edit type example shows part of the text added in the first or second edit respectively. The *Post ID* is the StackOverflow ID. Note that revisions of post with ID `postID` can be accessed via *http://stackoverflow.com/posts/`postID`/revisions*.

$R^{i_0}_{t_{a_0}}$, i.e. the original question posted at time t_{a_0}. Revision IDs are sorted according to time, each subsequent revision is an edit of the previous revision.

Users (the asker as well as anybody else) can also *comment* on a question or *answer* it. Let $C^i_{t_j}$ be a comment on question Q_i or any of its revisions at time t_j. Similarly, let $A^i_{t_k}$ be an answer to question Q_i (or any of its revisions) at time t_k. Which revision the comment or answer are referring to, depends on the timestamp of the comment or answer. We exploit these comments and answers and extract all pairs of original & edited question, with the following sequence characteristics:

$$R^{i_0}_{t_{a_0}} \rightarrow C^i_{t_j} \rightarrow R^{i_1}_{t_{a_1}} \rightarrow A^i_{t_k} \tag{5.1}$$

where $t_{a_0} < t_j < t_{a_1} < t_k$. The idea is to be able to automatically catch edits stimulated by discussions with the community.

Intuitively, we consider edits that:

- have been made potentially in response to a first comment; and
- after the edit, triggered the posting of an answer.

To further ensure that the edits occurred in response to the posted comment, we only consider those pairs of original and edited questions where there is some overlap in terms between the comment and the added text in the edit.

As an example, in response to a comment:

"Please add some source code"

a user might edit a question and add:

"My code: `[actual code]`*."*

With this basic filtering step we were able to capture around 170K quality-enhancing edits. The resulting question-edit pairs were then ranked according to the amount of editing, measured by the number of characters changed in the edited and original version of the question.

Our *non-edited questions* were selected from among all questions that were never edited and have received at least one answer. We ranked the non-edited questions according to their number of received answers – intuitively, the more answers a question receives, the higher is the engagement of community members with the question.

Extracting Edit Types Based on the categories identified in Section 5.3.1, we conducted a follow-up annotation study on 1000 *edited* questions randomly selected from the 25K most edited questions (i.e. those with the longest edits), with the purpose to derive labelled data for our edit types classifiers.

We collected annotations[9] for the questions according to four categories derived from our initial findings presented in Section 5.3.1: *Code*, *SEC* (merging the categories *Problem Statement*, *Example* and *Context*), *Attempt* (merging the *Solution* and *Attempt* categories) and *Detail*. The decision to group the categories as presented was taken due to the practical difficulties the annotators encountered deciding between them. In later stages, we discarded the *Detail* category due to the small number of annotated instances. Edits which do not fall into one of our categories were labelled as a "null edit".

We note, that for every question to be annotated, *all* edits of that question were labelled, i.e. $R^i_{t_j}$ for $j = 1...n$.

The annotations were then used to train three binary classifiers aimed at providing suggestions about the type of edit to be performed, for those questions that were deemed as in need for edits.

5.3.3 Hypotheses

This section presents the research hypotheses, based on the research questions posed in Section 5.1, we investigate in our work.

- **Hypothesis 1**: Communities attracting beginner's programmers (e.g. Android programming, Web design) receive a larger number of edited questions than communities which require more in-depth knowledge (e.g. Assembler programming, functional programming).
- **Hypothesis 2**: Users new to StackOverflow post questions in need of refinement. Over time, users learn how to post good quality questions.
- **Hypothesis 3**: Not only the time a user has spent on the portal is important, but also the amount of knowledge the user already has about a particular topic. We posit that users with substantial knowledge on a particular topic are less likely to post questions which require a substantial edit.
- **Hypothesis 4**: As the StackOverflow platform gained popularity, less and less questions requiring a substantial edit have been posted. Users

[9] We describe the annotation process in greater detail in Section 5.4.2.

read the guidelines and "learn" from different forums/portals how to properly ask questions.

- **Hypothesis 5**: New users are most likely to "forget" to add source code and previous attempts to their questions.

	#Questions Overall	#Edited Questions	#Non-edited Questions
Test: Extreme	14,920	7,460	7,460
Test: Confident	85,072	42,536	42,536
Test: Ambiguous	1,772,649	522,874	1,249,775
Training: Extreme	35,892	17,946	17,946

Table 5.2: Basic statistics of our training and test data for the edit prediction task. Since more non-edited than edited questions exist, for the *Extreme* and *Confident* partitions, the number of non-edited questions was matched to the number of edited questions by sampling a subset of all questions in the respective dataset.

5.4 Experimental Setup

We use the public StackOverflow dump[10]. Manual annotations, training and test data used in ours experiments are available for download at https://github.com/WISDelft/WIS_HT_2014. We consider, for training purposes, all questions posted up to and including December 31, 2012; the test set includes all questions posted between January 1, 2013 and September 6, 2013. We use a logistic regression-based classifier[11]. The feature set is composed of unigrams (terms) extracted from the dataset, an approach that has been shown to perform well for different prediction tasks in the past. The chosen classifier, though likely to not yield the best possible accuracy, allows us to gain valuable insights into the importance of different features.

5.4.1 Edit Prediction

The training and evaluation of the edit prediction classifier has been performed using the ranked list of edited and non-edited questions described in Section 5.3.2.

[10] Available online at https://archive.org/details/stackexchange
[11] Implemented in sklearn http://scikit-learn.org

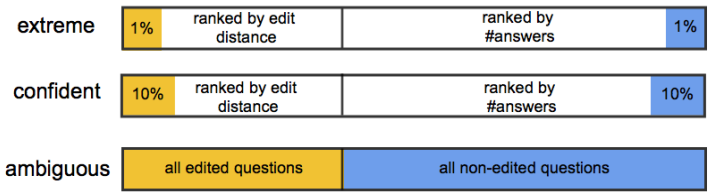


Figure 5.1: Both the training and test data were partitioned in three ways. The edit prediction classifier was trained on the *Extreme* set of the training data. The evaluation was performed on all data partitions of the test data.

Given these two rankings of the questions in the positive (*edited*) and negative (*non-edited*) class, we create three different data partitions, presented in Figure 5.1.

- The **Extreme** set contains the top 1% of positive and negative samples.
- The **Confident** set contains the 10% highest ranked edited and non-edited questions respectively.
- The **Ambiguous** set contains all edited as well as all non-edited questions.

We derive this partitioning of the data separately for our training and test data. We train our edit prediction classifier on the *Extreme* data partition of the training data (i.e. questions posted until the end of the year 2012) and evaluate the performance of the classifier on the *Extreme*, *Confident* and *Ambiguous* data partitions of our test data (questions posted in 2013).

For training purposes, due to the skewedness of the class distribution (there are more non-edited than edited questions), we randomly sample from the negative class until we have reached the same number of samples as exist in the positive class. A similar sampling process is also used for the test data, with the exception of the *Ambiguous* set, which includes all test questions.

The reason for experimenting with different data partitions is the nature of the task. Our overall goal is to predict for each and every question in our test set whether or not it requires an edit. Due to the nature of the questions, we expect that questions in the *Extreme* test set can be classified with a higher accuracy than questions in the *Ambiguous* test set.

Table 10.1 contains an overview of the total number of questions used for training and test purposes. We train on nearly 36,000 questions and test our pipeline on up to 1.8 million questions.

5.4.2 Predicting the Edit Type

Given a question which has been flagged as "to edit" in the first step, this processing step determines which aspect(s) of the question require an edit.

The 1000 annotated questions feature an average of 3.05 ± 1.84 edits. Three trusted annotators evaluated disjoint sets of 300 questions each. Additionally, a common set of 100 questions were labelled by all three annotators to test the agreement. The inter-annotator agreements for the four edit categories are shown in Table 5.3.

Edit Type	Code	SEC	Detail	Attempt
Kappa	0.67	0.59	0.19	0.65

Table 5.3: Inter-annotator agreement of edit category annotation, measured by Fleiss' Kappa.

The number of questions belonging to each category are reported in Figure 5.2. We used a majority consensus approach to determine the category of the 100 overlapping questions. Recall, that we annotate every edit of a question, and thus the total number of items shown in Figure 5.2 exceeds 1000. Of all edits, 30.75% could not be assigned to any of the four categories. We did not observe significant differences between the edit type distribution at different edit iterations (i.e. first edits are similarly distributed to second or third order edits).

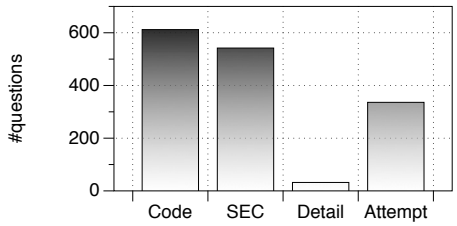


Figure 5.2: Annotation study results: number of questions with an edit from a particular category. The SEC category captures the problem **S**tatement, **E**xamples and the **C**ontext.

We observe that *Code*, *SEC* and *Attempt* are often occurring categories, indeed more than half of the questions have at least one *Code* edit (it is also not uncommon to have several). For these three categories the inter-annotator agreement is also moderate to high (0.59 or higher). In contrast, the category *Detail* suffers both from very low inter-annotator agreement and few positive annotation results.

We train three binary classifiers, dropping the *Detail* category from further experiments due to the annotator disagreement and the small sample size. All questions with a particular edit type belong to the positive class for that edit type classifier, the remaining questions of our annotation set form the negative class. The classifier training follows a similar setup to step one. We derive features from the original question and include it in the training set for a classifier if at least one of the question's edit was annotated as belonging to the classifier's category. Due to the small size of the training data though we cannot rely on word unigrams as features. To avoid overfitting, we employ Latent Semantic Analysis [59] and rely on the 100 most significant dimensions as features. To evaluate the edit type prediction task, we use 5-fold cross validation.

5.5 Experiments

We first present the results of our edit and edit type prediction tasks. Subsequently we present an analysis of a number of user-dependent factors that we hypothesize to influence the likelihood of a posted question requiring an edit (based on the hypotheses presented in Section 5.3.3).

5.5.1 Edit Prediction

The performance of our classifier on our test sets is presented in Table 5.4. As expected, the best results are achieved for the *Extreme* test set with an F1 score of 0.7. The recall of 0.78 implies that most questions which require an edit are classified as such by our approach, thus clearly demonstrating its feasibility. The classifier is trained on a feature set with a total of 7,206 features.

Test type	Precision	Recall	F1
Extreme	0.63	0.78	0.70
Confident	0.58	0.69	0.63
Ambiguous	0.51	0.65	0.57

Table 5.4: Classifier performance on the edit prediction task across our three test sets.

When comparing the performance of *Extreme* and *Ambiguous*, the impact of the test set generation process becomes evident. For the *Ambiguous* test set the performance of all three measures drops significantly. This is not

surprising, as the middle ground questions (containing small edits or being poorly phrased but remaining unedited) are the most difficult for a classifier to identify correctly. We conclude that our proposed classifier, if employed on the stream of new StackOverflow questions, would be able to spot the most severe cases of questions requiring an edit with high accuracy. We leave the exploitation of more advanced machine learning models and additional features for future work.

Important Features One of the benefits of a regression-based classifier is the ability to gain insights about the importance of different features based on the feature coefficients. In Table 5 we list the features (unigrams) with the highest and lowest coefficients respectively (after feature normalization). For instance, the term *microsoft* is an important feature for to-be-edited questions, while *lexer* is negatively associated with question edits, presumably because users discussing lexers have specific problems and a relatively deep understanding of their topic.

Unigram	**Coef.**	**Unigram**	**Coef.**
dbcontext	0.88	mental	-0.29
microsoft	0.57	nicer	-0.31
xx	0.57	understood	-0.31
com	0.55	pre-compil	-0.34
tick	0.47	lexer	-0.41
neater	0.46	c/c++	-0.42
byte	0.45	firstnam	-0.47
inbuilt	0.44	testabl	-0.53
socket	0.42	string	-18.48
reproduc	0.39	archiv	-19.94

Table 5.5: Regression coefficients of the most positively and negatively weighted features (unigrams) for the edit predictiont ask.

5.5.2 Edit Type Prediction

We now consider step 2 of our pipeline - the prediction of the type of edit(s) required to create a well-formed question. The results are shown in Table 5.6, rows one to three.

While the edits of *Code* and *SEC* can be predicted with moderate to high accuracy, the prediction of the *Attempt* category is essentially random.

Strategy	Edit category	Nr. positive	Nr. negative	Precision	Recall	F1
No augmentation	Code	612	388	0.63	0.83	0.71
	SEC	542	458	0.57	0.62	**0.59**
	Attempt	336	664	0.39	0.45	0.40
Positive augmentation	Code	8157	338	0.63	0.92	0.75
	SEC	542	458	0.57	0.62	0.59
	Attempt	2387	664	0.40	0.49	0.44
Positive+ negative augmentation	Code	8157	8157	0.63	0.95	**0.76**
	SEC	542	542	0.55	0.49	0.52
	Attempt	2387	2369	0.38	0.56	**0.45**

Table 5.6: Classifier performance on the edit type prediction task. Numbers underlined are the ones higher than previous classification version. The best F1 scores in all edit type prediction tasks are highlighted in bold. Note that Nr. positive and Nr. negative only indicates the number of questions that affect training of the classifier. Precison, Recall and F1 are calculated based on the 1000 annotated questions.

Automatically Augmenting the Training Data Having so far relied on our manually annotated data only, we now turn to an automatic approach to augment the training data (the test data is fixed to our manually annotated questions). The goal is to provide sounder evidence on the performance of our predictors. We test two augmentation strategies:

1. **Positive augmentation**: we assume that questions with the term `code` appearing in the edited version while not in the original version have a big chance to be a positive question of edit type *Code*; this is verified in our annotated dataset where this is true for more than 38% of the questions in the edit type *Code* category. We use this strategy to collect additional training data from the *Extreme* training set; for the edit type *Code* we identified nearly 7000 additional questions. We followed the same approach for the *Attempt* category, relying on the term `tried` (this assumption holds true for 21% of our annotated data set). No augmentation was performed for category *SEC*, as no indicative terms could be determined.

2. **Negative augmentation**: We consider non-edited question in the *Extreme* training set as well-formed questions, and include similar number as edited questions to be the instances of the negative class.

To ensure that the classification results are not influenced by our selection criteria, the features `code` and `tried` are removed in the training phase.

The classifier performance with both types of enlarged training data are reported in Table 5.6, rows four to nine. In the case of positive augmentation it can be observed that both the *Code* and *Attempt* prediction performances increase. The improvements in F1 stem from an increase in recall. This is natural since the augmented training data contains only positive questions.

After negative questions were added as well, the edit type predictions *Code* and *Attempt* are very slightly enhanced. This indicates that the negative questions does not contain much information of each other. For type *SEC* the classifier performs as poorly as a random baseline.

To summarize, we have found that the edit prediction task can be solved with high accuracy, while the edit type prediction task is more difficult to solve. We have presented strategies to semi-automatically enlarge the training data which have been shown to be beneficial for the *Code* and *Attempt* categories.

5.5.3 Hypotheses Testing

We now turn to an analysis of our hypotheses presented in Section 5.3.3.

Up to now we have only considered the question content in edit and edit type prediction. We now explore the impact that different factors can have on the quality of a question. Such factors include the topic of a question, the user's prior experience on StackOverflow, user knowledge on the question's topic, and the temporal influence of StackOverflow. We first test our **hypotheses H1-H5**, then add related features for the prediction tasks to our classifier to investigate whether they can make a difference.

Topical Influence. We investigate **hypothesis H1**, i.e. if questions about particular frameworks or languages (e.g. `JavaScript`, `Java`), in particular those often used by programming beginners, are more prone to requiring an edit than questions related to more advanced topics such as software engineering (e.g. `design-patterns` or `compilers`).

For simplicity, we consider the tags assigned to each question as indicator of a question's topic. To avoid the influence of insignificant edits, we consider all questions of the *Confident* datasets (both training and test). Since a question may be assigned multiple tags, a question may appear in multiple tag sets. We rank the tags according to:

$$\frac{\#\textit{questions with substantial edits}}{\#\textit{questions without an edit}} \tag{5.2}$$

filtering out all those tags that appear too infrequently in the data set. We consider this ranking to provide us with an indication of a community's amount of beginners.

Rank	Tag	Ratio	#Questions in *Confident*
1	asp.net-mvc-4	6.16	505
2	jsf	6.02	615
3	symfony2	5.57	338
4	r	4.34	2,067
5	opencv	4.10	402
6	matlab	4.02	981
7	core-data	3.91	446
8	angularjs	3.67	288
9	mod-rewrite	3.52	297
10	asp.net-mvc-3	3.50	1,443
	....		
192	vim	0.52	746
193	visual-studio-2008	0.50	921
194	web-applications	0.49	774
195	oop	0.45	2,711
196	database-design	0.45	1,220
197	unit-testing	0.44	1,526
198	logging	0.44	624
199	testing	0.41	849
200	design	0.34	1,386
201	svn	0.27	1,186

Table 5.7: Overview of the topics (tags) which contain the most and least edited questions. All available data was used to generate the rank and ratios. The last column shows the number of questions in the *Confident* data set.

Table 5.7 provides an overview of the ten most and least edited topics (identified by their tags) in our data set. As hypothesized, the top-ranking topics are those more framework or language related, while low-ranking topics are more generic or advanced. For instance, `asp.net` questions usually require a lot of edits. In contrast, topics like `design` or `testing` require edits with a considerably lower likelihood.

We also report the number of questions a tag is assigned to in the *Confident* data set. It can be observed that the tags of most edited questions usually occur less than the non-edited ones (except the `r` tag). This indicates that not the large number of beginners leads to poorly phrased questions. It is more likely that these questions need to be edited because they are more complex and require more clarifications.

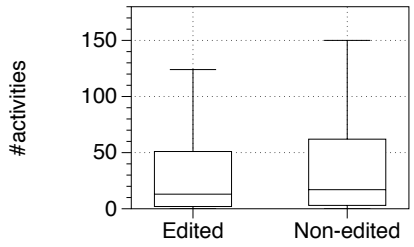


Figure 5.3: Influence of user experience on posting a question which requires an edit.

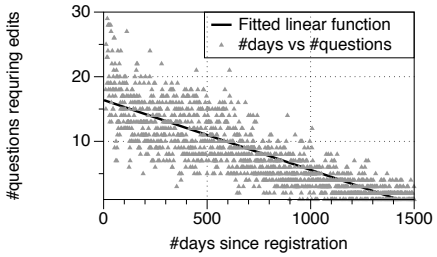


Figure 5.4: Influence of user knowledge on question edits. Results shown for topic (tag) `C#`.

User Influence. Hypothesis H2 is concerned with the user effect - how does a user's familiarity with the portal StackOverflow affect the probability of an edit? If **hypothesis H2** holds, we expect that the probability of a substantial edit decreases with increasing user experience with the platform. Such experience can be implied based on different types of user actions such as posting questions, answering, commenting or voting on postings.

We use the *Confident* data set (training & test), which contains a total of 151,762 users – (16.4%) of all StackOverflow askers. For each question, we determine the number of questions and answers in the entire data set (not limited to *Confident*) the asker has posted previously, then bin them into two groups: edited vs. non-edited questions. The comparison of these two groups is shown in Figure 5.3 in the form of a box plot. The number of past activities of a user is - as hypothesized - a significant indicator for the likelihood of a question edit. Users with fewer activities are more likely to edit their questions than more experienced users (to a statistical significant degree, p-value$<$0.001 by a Mann-Whitney test).

Knowledge Influence. Hypothesis H3 considers not only the activity of a user in the past (regardless of the topic), but also the knowledge of a user on a topic. In particular, we hypothesize that the number of questions requiring an edit decreases as a user gathers more experience on the topic (as she becomes more familiar with the terminology, etc.).

To evaluate this hypothesis, for each asker in the *Confident* data set (training+test) we plot the number of days since registering on StackOverflow vs. the number of specific topic-related questions that require a substantial edit asked on this topic. As before, we use tags as topic indicator.

Our analysis shows that these two variables are highly negatively correlated, with a Spearman correlation of -0.72 (p-value$<$0.001). We remove all

users with a registration date older than 1500 days, and denote the activity of a user by a vector $(a_1, \ldots, a_{1500})$ where a_i denotes the number of questions and answers posted by this user at day i since his registration. Figure 5.4 shows the cumulative vector for all users involved in the topic `C#`. It can be observed that as time passes, a user asks less questions that require substantial edits. Though we only present the results for `C#`, we note that we observe the same trends for the top 20 topics (tags) on StackOverflow, which include `Java`, `iOS` and `Python`.

Temporal Influence. Similarly to **hypotheses H2** and **H3**, we can also evaluate **H4** by considering all questions posted in a particular year. If **H4** holds, we expect to see a decreasing trend in questions requiring an edit. There is an influential factor, though, which will lead to more questions that require edits: new users registering and asking questions. Figure 5.5 plots:

$$E = \#\textit{edited questions} - \#\textit{non-edited questions}$$

in the *Confident* data partition over time, while Figure 5.6 depicts the evolution of user registrations in the same time period.

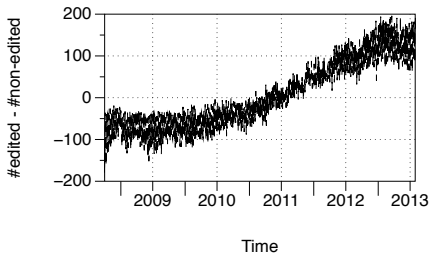


Figure 5.5: Increase in edited questions over time.

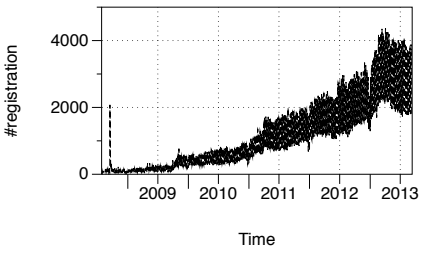


Figure 5.6: Increase in user registration over time.

The Spearman correlation between E and the number of user registrations is 0.79 with a p-value$<$0.001. This result provides additional support to the motivations of our work, as it shows that, despite the fact that an individual user asks fewer questions when he stays longer on StackOverflow, the increasing popularity of the platform leads to the creation of several more questions that could benefit from a systematic assessment of their quality.

Influence of User "Age" on Edit Type. Hypothesis 5 is concerned with the role that user seniority plays in influencing the types of information (*Code*, *Attempt*, or *SEC*) that are (not) initially included in the questions.

For each of the 1000 annotated questions, we calculate the age of the question as the difference between its posting date and the registration date, in StackOverflow, of its asker.

Figure 5.7 depicts the difference, in terms of age, of edited and non-edited questions in the context of the *Code* edit type: we observe that this type of edits is significantly (p-value<0.001 by a Mann-Whitney test) more likely to occur in the early days of a user's activity on the platform; *SEC* and *Attempt* edits do not show significant differences.

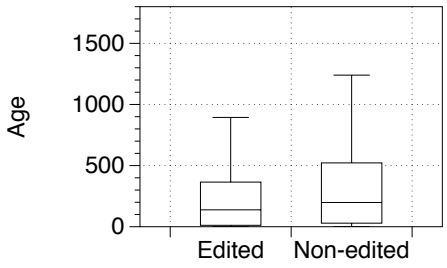


Figure 5.7: Influence of user age on posting a question which requires a *Code* type edit.

Influence on Prediction. In a final experiment, we created additional features for edit and edit type prediction based on the results of the investigated hypotheses. The following features were added to the existing feature set: 1) tags of a question, 2) #activities of the asker, 3) #days between the registration of the asker and the time she posted the question, and, 4) #days between a question was posted and the time StackOverflow was launched.

In our experiments we did not observe substantial differences in F1 when adding those features to our original (unigram-based) feature set. This indicates that the content, i.e., the terms in a question, are more important that contextual factors for predicting the question (type) edit.

5.6 Conclusion

As CQA systems grow in popularity and adoption, the ability to provide automated quality enhancement tools is a key factor to guarantee usability, reliability, and high knowledge creation quality. In this chapter we explored a specific aspect of user contributions: the formulation of well-formulated questions. In order to receive useful answers, a question should feature positive characteristics such as specificity (i.e. provide enough details to understand

the nature of the problem), and clarity (i.e. provide examples, or personal experiences).

We analysed the editing behaviour of StackOverflow users, and identified three main classes of useful editing actions. We then applied machine learning techniques to define an approach for the automatic suggestion of edit types for newly created questions. With respect to the research questions listed in Section 5.1 we can draw the following conclusions:

- **RQ1**: Question edits are a very good indicator of the quality of a given question, as their presence is also a reflection of several distinct traits of the asker (e.g. being new to a given technology, knowledge in the targeted topic, etc.).
- **RQ2**: Using a simple unigram model, we observe classification accuracies (F1) between 63% and 70%. This is a very promising result which indicates the possibility for significant improvements when adopting more sophisticated techniques.
- **RQ3**: Out of three identified classes of edits, only one (namely *code refinement*) features good prediction performance. The results are encouraging, but suggest that a more in-depth analysis of the different type of editing actions is required, to gain a better understanding of their features.

Chapter 6

Modeling Task Complexity in Crowdsourcing

This chapter studies task complexity from the point of view of crowd perception. By collecting worker assessment on task complexity in human computation systems, we study how task complexity is perceived among workers, to look for related properties that are consistently deemed across workers. We then introduce a set of task design features (metadata, content, and visual) and a high-dimensional regression method for measuring task complexity. Finally, we demonstrate the importance of task complexity, by analyzing its role in improving the prediction of task completion rates.

This chapter is published as "Modeling Task Complexity in Crowdsourcing" [232], by J. Yang, J. Redi, G. Demartini, and A. Bozzon in Proceedings of the 4th AAAI Conference on Human Computation and Crowdsourcing, pages 149-158. AAAI, 2016.

6.1 Introduction

Tasks play a significant role for all the actors involved in crowdsourcing campaigns: their designers – i.e. the *requesters*; their executors – i.e. the *workers*; and the *platform* (e.g., Amazon mTurk), which hosts them and enables their discovery, assignment, and reward.

Studying task properties is key for addressing core crowdsourcing problems such as task assignment and optimization, and worker retention [227]. To this end, previous research extensively covered the study of task metaproperties, e.g., setting appropriate rewards [43], with the goal of improving the effectiveness of human computation algorithms, and the quality of crowdsourcing output in general. *Time* is the dimension along which the human cost (e.g., fatigue, stress) associated with the execution of crowdsourcing tasks has been typically quantified. Incentives such as monetary rewards strongly depend on the estimation of task completion time. *Complexity*, instead, reflects not only temporal demands of task execution, but more holistically the real cognitive effort that performers need to put into the completion of tasks. Interestingly, this task property has so far been mostly neglected by crowdsourcing research.

Objective. *Complexity* has been identified as one of the most important task properties in a variety of fields studying the relationship between humans and computer artifacts. It is a construct that is widely used in behavioural sciences, towards investigating task cognitive load; nevertheless, it is surprisingly difficult to describe. To advance the theory and practice of crowdsourcing, we advocate the need for a systematic examination of how task complexity is perceived by workers, what contributes to task complexity, and of how it affects execution performance and output quality.

Being able to measure and predict task complexity can be highly beneficial for both workers and requesters. Example applications include more accurate reward estimation for a given task, better task routing approaches that take into account worker skills and expertise, and simpler worker reputation management by letting workers decide at which level of complexity to work without risks of compromising their reputation (e.g., a worker may want to start working on tasks with low complexity and increase it as she becomes more confident). Crowd answer aggregation [218, 196] and evaluation of worker performance [110, 111], where the complexity of the completed task can be taken into account when assessing work quality, may also benefit from it. Finally, a better understanding of task complexity dimensions may also inform better governance within enterprise crowdsourcing (i.e., tasks crowd-

sourcing within an enterprise to the crowd consisting of company employees) where different sociological issues exist (e.g., workload balance, different reward schemes, etc.) [219].

Original Contributions. This chapter makes a concrete step towards advancing our understanding of the complexity of crowdsourcing tasks and its quantification. We borrow from one of the most widely-used task complexity taxonomies [36], focusing on task complexity as a property of tasks perceived by workers during the interactions between them, i.e., subjective task complexity. Specifically, we seek answers to the following research questions:

- **RQ1**: How is the complexity of crowdsourcing tasks perceived by workers, and distributed over tasks?
- **RQ2**: Which task features can characterize crowdsourcing task complexity, and based on them how to predict task complexity in an automatic way?
- **RQ3**: How do task complexity features affect task performance?

By re-instantiating 61 tasks published in Amazon mTurk, we collect subjective complexity evaluations from workers by using the NASA Task Load Index. We define a quantifiable measure of subjective complexity (i.e., Mean Complexity Score), which shows that task complexity is coherently perceived by different workers, and which we show to be significantly influenced by task type. To understand the influence of task design on complexity perception and to ultimately develop an automatic way for measuring task complexity, we propose a high-dimensional regression method, named `MFLR`, to model the key dimensions of complexity from three classes of *structural* features of a crowdsourcing task, including metadata, semantic content, and visual appearance. We show that both the visual appearance and language used in task description can largely influence the perception of task complexity, and thus can be used to measure it. Then, we demonstrate that the same feature set can be used to improve task throughput prediction. We show that complexity is a relevant task modeling property, thus contributing a better understanding of how task formulation can affect task execution performance in crowdsourcing markets.

6.2 Related Work

Task Complexity. Complexity is a multifaceted property of human-executed tasks, for which it is difficult to establish a holistic definition. It is generally agreed that *task complexity* [36, 225] depends on both objective task properties and individual characteristics of the task doer. As such, task complexity can be operationalized in terms of: 1) intrinsic complexity, which does not account for subjective perception and individual differences of task doers, but focuses exclusively on the features of the task; 2) subjective complexity, which measures the task as a function of the perception and handling of its performers; and 3) relative complexity, which considers the relative relationship between the difficulty of the task and the capabilities of the task doers. Subjective and intrinsic task complexity are related, as the former can be used to explain how the latter is handled by workers. In a comprehensive review of literature in information systems and organizational behavior, [36] concluded that "any objective task characteristic that implies an increase in information load, information diversity, or rate of information change can be considered a contributor to complexity". In this chapter we focus on *subjective* complexity of crowdsourcing tasks, aiming at its quantification and the identification of task features that contribute to it.

Measuring Complexity of Crowdsourcing Tasks. Previous work investigated the relationship between task length and the trade-off between execution speed and result quality, showing how decomposition into simpler micro-tasks might lead to slower completion time but increased accuracy [44]. This recent result indicates even further the need for understanding the impact of task complexity on the dynamics of a crowdsourcing marketplace and how this affects all actors involved. While task complexity has been used in other work on crowdsourcing – e.g., by means of a post-task questionnaire [44] or for the purpose of measuring worker effort and motivation [186], our work is the first one looking at the different factors affecting perceived task complexity and to propose and evaluate methods to predict it given task properties.

[43] propose a method to quantify the *effort* involved in task execution, meant to be used to estimate the appropriate monetary rewards for a given task. The approach is based on a pilot launch of the task, within which the time available for task completion is selectively manipulated. The approach is able to estimate the time needed to obtain completion while achieving a given output accuracy level. W.r.t. [43], our work aims at measuring task complexity beyond completion time, and in a fully automated way, i.e.,

avoiding the need for a task to be "piloted" in order to estimate its required effort. We use a standard method to measure the ground truth perceived complexity of a task – that is the NASA Task Load Index (`NASA-TLX`) – and instrument a set of features (i.e., metadata, content, and visual, fully computable from the task data and html code) to predict task complexity.

`NASA-TLX` finds application in a variety of experimental tasks [92], including the domains of information retrieval [88], Human Computer Interaction [136]. Recently, it has been also used in crowdsourcing, as a tool to assess subjective judgment of task difficulty [157], or to support the development of semi-automatic reward strategies [43]. The focus of our work is on reaching a better understanding of the different dimensions involved in measuring the subjective complexity of a crowdsourcing task. This will allow the creation of tools and quantitative measures to support worker and requester interaction, and it paves the way for novel research focusing on crowdsourcing of complex tasks [122].

Micro-task Crowdsourcing Markets. Micro-task crowdsourcing platforms are becoming more and more popular for both academic and commercial use. A variety of tasks is being crowdsourced over these platforms, with workers executing them in exchange of a small monetary reward. A taxonomy of popular micro-task type has been proposed by [70], with audio transcription and survey being identified as two of the most popular task type [63]. In our work we use the same taxonomy to characterize tasks first and verify the impact of task type on complexity later.

6.3 Measuring and Modeling Task Complexity

To enable more efficient and fair mechanisms for microtask execution, we need a better understanding of which task features influence the success and effectiveness of online work. To this end, it is crucial to quantify the complexity of the tasks that humans (workers) have to carry out through (or in collaboration with) computer systems.

6.3.1 Measuring Complexity with NASA TLX

In literature, hardly any characterization of crowdsourcing task complexity has been explored. We introduce here methods for characterizing and quantifying subjective complexity towards answering **RQ1**.

Subjective complexity is defined in terms of workers' experience with a task [36]. It relates to the notion of workload, since it is defined as the perception of the level of complexity associated with the performance of a task. Subjective complexity is related to intrinsic complexity, but they are not necessarily identical. Intuitively, tasks of a given intrinsic complexity might be more demanding for a worker than another. Also, tasks can require high effort (high subjective complexity) without necessarily being structurally more complex (arguably, a text summarization task is more demanding than an image annotation one), but other tasks might be demanding due to their structural complexity (e.g., a long survey with many radio buttons).

In the ergonomics literature, many instruments exist to measure workload, mostly based on self-assessment. Among them, the NASA Task Load Index (`NASA-TLX`) [93] is the most widely adopted. `NASA-TLX` is "a multidimensional rating procedure that provides an overall workload score based on a weighted average of ratings on six sub-scales: Mental Demands, Physical Demands, Temporal Demands, Own Performance, Effort and Frustration". The test is straightforward: after completing the task to be evaluated, subjects are asked to rate workload along each of the subscales (typically, through 20-point likert scales ranging from 0 to 100, with endpoints anchored "low" and "high"). Then, subjects perform a pairwise comparison of the subscales, chosing among a pair the subscale which contributes the most to the overall workload. The pairwise comparison provides the relative weighting for each subscale [93]in the final overall `NASA-TLX` score, which is then determined as the weighted sum of each subscale score multiplied by the corresponding weight. In this work we adopt the resulting `TLX` score, ranging between 0 and 100, to measure subjective complexity of crowdsourcing tasks.

6.3.2 Modeling Complexity with Task Features

Our second research question (**RQ2**) concerns determining which *quantifiable* and *immutable* properties of a task can be used to estimate subjective complexity. The focus is on properties that relate to the task structure (e.g., instantiation properties, graphical layout, instructions) but not to its matter. The difference is subtle but significant. A task can be instantiated having the same structure (e.g. GUI and instructions), but different content. For example, the same task structure could be instantiated to ask workers to search for a bird in a picture. In this case, the specific picture proposed to the worker for the search would be the matter. Presenting the worker with a realistic picture of a bird rather than Miro's "The Migratory Bird" may significantly alter the complexity of the task. Investigating the influence of task matter

on crowdsourcing task complexity poses issues of (1) data retrievability and (2) (multimedia) content interpretation, which is beyond the scope of this chapter. In the following we explore three classes of task structural properties, and specifically: *metadata* features, *content features*, and *visual features*. We provide a brief introduction to each category, and refer the reader to the companion page for a full description of the feature set[2].

Metadata Features include task attributes defined at task creation time. They are used by a requester to provide an overview of the activities to be performed (i.e. `Title` and `Description` length), the required qualifications (e.g. worker `Location` and minimum `Approval Rate`), the estimated `Allotted Time` time and `reward`, and the number of `Initial Hits`. Our hypothesis is that metadata features reflect complexity as seen from the requester's point of view: intuitively, the task requester would reward higher compensation for more complex tasks, or provide a longer description to attract suitable workers.

Content Features aim at capturing the semantic richness of a task. Research in related domains – e.g. community question-answering systems [228], website complexity [106] – suggests that content features could be indicative of the requirements and quality of task specification, which could in turn influence the perceived complexity of a task. Content features include numerical properties like the `Amount` of `Words`, `Links`, and `Images` in the task body, but also the `Keywords` used to concisely describe the task; the actual words (`unigrams`) contained in its title, description, and body; and the high-level `Topics`, extracted from task title and content using Latent Dirichlet Allocation (LDA) [22]. We hypothesize that the content of a task captures its requirements and hence its complexity. Also, the use of language features like unigrams can help highlight poorly formulated task instructions, which may increase workers' cognitive load at execution time.

Visual Features relate to visual properties of a task, including its layout and color palette. As the visual complexity of a Web page is known to influence the users' cognitive effort during interaction [91], as well as users' aesthetics impression [178], we hypothesize that the look and feel of a crowdsourcing task can also have influence on its complexity. We study visual features that insist on (1) the general Web page structure [106] such as `Body Text Percentage`, amount `stylesheet` and `script` files, and (2) the `visual areas` [178] such as number of `Text groups` and `Image areas`[3]. Intuitively,

[2]https://sites.google.com/site/hcomp2016complexity/

[3]The code used to extract visual areas is available at http://iis.seas.harvard.edu/resources/aesthetics-chi13/

Task Type	Count	Percentage
Survey (SU)	4	6.6%
Content Creation (CC)	19	31.15%
Content Access (CA)	4	6.6%
Interpretation and Analysis (IA)	17	27.87%
Verification and Validation (VV)	2	3.3%
Information Finding (IF)	14	22.95%
Other	1	1.6%

Table 6.1: Distribution of task type in dataset.

one could expect features like the amount of text groups to be positively correlated with task complexity, while some others such as the amount of `Emphasized Body Text Percentage` to be negatively correlated with task complexity – a higher emphasized body text percentage may suggest that the task designer has put effort in managing the user experience. We also study color-related features, previously investigated on images [94] and websites [178]. `Colorfulness` is considered as one of the most notable features that influence emotional reaction [51]. We therefore hypothesize that it could also influence worker's mood during task execution, and thus influence their perception of task complexity. The color features we will study include the `histogram` of colors, and `HSV values`.

6.4 Is Complexity Coherently Perceived by Workers?

Our study started with an experiment where we collected task complexity evaluation from workers. This section introduces our experiment and the analysis of how task complexity is perceived by workers and distributed over task type.

6.4.1 Experiment

Setup. To collect subjective complexity data, we first created a dataset of real mTurk tasks, to be then re-instantiated, executed, and evaluated by workers. As a first step, we extended *mTurk-tracker*[4] to enable the retrieval of the complete set of resources associated with a task posted on mTurk, including meta-data, formatting (e.g. Javascript, CSS) and, when publicly

[4]http://www.mTurk-tracker.com/

available, content (e.g. images, audios). We performed the crawling during the first week of August 2015, retrieving 5487 tasks. We composed our final dataset of 61 tasks by picking from each requester active in the observation week one HIT per task type. The considered tasks have been labeled by the task type classifier developed in [63] for the task type taxonomy proposed in [70]. This was necessary because HITs from the same requester are often similar to each other, and the majority of HITs are posted by a small number of requesters. Selecting one HIT per task type and requester guaranteed diversity in the task set in terms of both HIT content and visual presentation, as well as in task type; an overview of the composition of the dataset is given in Table 6.1. We then instantiated the 61 tasks based on the crawled task data. To minimize the chance of learning bias, we use another platform, i.e. Crowdflower, to collect the complexity evaluation for these tasks. We turned to the CrowdFlower crowd also to include in the evaluation the judgment of workers from a broader set of countries, thus reducing the risk of country-specific biases. Then, we appended at the end of each task the `NASA-TLX` questionnaire of task complexity assessment, and we asked workers to fill it in with respect to the task they just executed. Workers were recruited among Crowdflower Level 3 contributors, and tasks and `TLX` completion were performed in an externally hosted server.

To obtain reliable complexity assessment, we aimed at having 15 workers executing each task and completing the related `TLX`. In addition, to control for the quality of the complexity evaluations, we implemented a post-hoc task execution filtering mechanism. Due to lack of supervision during task execution in crowdsourcing, misunderstandings of the task instructions, as well as low engagement or commitment to the task, unreliable task outputs are to be expected [99, 65]. As `TLX` entails subjective assessment of complexity, it was not possible to adopt a quality control based on golden answers [8]. Therefore, we implemented a mechanism of control for self-consistency. When presenting the workers with the 15 pairs of subscales at the end of `TLX` questionnaire, we repeated three pairs twice (non consecutively, so the workers would be less aware of the repetition). We would expect unreliable workers to skip through these pairs, randomly selecting an answer to finish the task fast, and possibly giving inconsistent answers on the repeated pairs.

Results. The 61 crowdsourcing tasks were executed and evaluated by 13 to 16 workers each ($M = 14.8 \pm 0.572$). In total, we obtained 903 evaluations, including: (1) a set of 6 judgments of complexity, each expressed along a different subscale, representing a different *complexity factor* (Mental, Physical, Temporal, Performance, Effort and Frustration); (2) a set of six *weights*, each

representing the relevance of each factor in computing the final task complexity and (3) an overall *task complexity score*. For each completed task, we also recorded both the time taken by each worker to complete it and the time taken to fill in the follow-up TLX questionnaire.

The filtering of task executions and evaluations was enacted when: 1) more than two mistakes were made in the control questions (i.e., repeated pairs of complexity factors in the questionnaire); and 2) the time taken to complete the TLX questionnaire was outside the range $MEAN\_CT(i) \pm 2 * std\_CT(i)$, where $MEAN\_CT(i)$ is the average across workers of the time taken to fill in the questionnaire for task i and $std\_CT(i)$ is the related standard deviation.[5] For 34 of the 903 TLX completions (3.8%), workers made mistakes in all control questions; in 107 cases (11.8%), 2 out of the three control questions were answered in a wrong way. As a result, 15.6% of the evaluations were discarded. Of the remaining 762, 52 took either too short or too long to be completed (5.8%); hence, the following analysis is based on the remaining 710 evaluations (11.64 ± 1.693 per task).

6.4.2 Perception and Distribution of Task Complexity

Inter-evaluator Agreement. The purpose of the subjective study was to establish a *Mean Complexity Score* (MCS) for each task, expressing the complexity of a given task as perceived, on average, by a crowd of workers executing it. This was functional to our follow-up experiment of complexity prediction. To establish MCS as ground truth, we verify the presence of a sufficient agreement among workers (evaluators) in expressing complexity scores: a too large disagreement would give large confidence intervals of the MCS, making them unreliable as ground truth.

Traditional inter-evaluator agreement measures are not applicable in our case, as they often assume repeated measures (e.g., Cronbach's alpha). Krippendorff's alpha does not have this limitation, but it has been shown to be of limited reliability for subjective evaluation tasks [8]. A possible different way to look at the problem is to check the extent to which individual evaluations are spread around the mean of the complexity (or complexity

[5] $MEAN\_CT$ was computed per each task separately, rather than across tasks. This is due to the presence of a significant effect of the task i on the TLX completion time (Kruskal-Wallis Test, $H = 142.89, p < 0.001$). This suggests that the characteristics of the task may have influenced the engagement of the worker with the task execution (and thereby questionnaire completion); workers poorly engaged with the task may have completed the TLX skipping through it (too fast) or doing other things while at it (too slow), either ways producing poorly reliable task evaluations.

Factor	Complexity	Mental	Physical	Temporal	Performance	Effort	Frustation
α	0.2785	0.2627	0.2745	0.2897	0.2507	0.2503	0.2937

Table 6.2: SOS Hypothesis α values for Mean Complexity Scores and Mean Factor Scores.

factor) value per task. This type of analysis, often used in visual quality assessment tasks, can be applied to any type of subjective evaluation involving a pool of participants scoring the same item (in our case, a task i), for which a mean opinion score along some dimensions (complexity and its factors, in our case) is sought. The SOS hypothesis [98] is a useful tool to perform such type of analysis. It stems from the observation that Mean Opinion Scores (MOS; in our case, mean complexity - or complexity factor - scores for the tasks) and the spread of the individual scores around the MOS (as measured by their Standard Deviation, or SOS) are linked by a squared relationship. Specifically, if the dependent variable (i.e., complexity) is measured on a K-point scale, with v_1 being the lowest value of the scale and v_K being the highest, for each task i we can define the relationship $SOS(i)^2 = \alpha(-MOS(i)^2 + (v_1 + v_K)MOS(i) - (v_1 * v_K))$, with $MOS(i)$ the Mean Opinion Score for task i and $SOS(i)$ the related standard deviation. The parameter α that regulates this relationship can be found by fitting the MOS to the SOS data. Its value can be compared with that of other subjective evaluation tasks, which are deemed to be more (high α) or less (low α) prone to high variability in evaluations.

Table 6.2 shows α values computed for complexity evaluations as well as evaluations of the individual complexity factors (mental, physical, temporal, performance, effort and frustration). α values range between 0.25 for the effort factors and performance and 0.29 for the frustration factors. Complexity evaluations have an α value of 0.28. This value is similar to what could be obtained in other subjective evaluations tasks (e.g. smoothness of web surfing, VoIP quality, cloud gaming quality, or image aesthetic appeal [98, 177], and we consider it acceptable. We thus conclude that subjective task complexity is coherently perceived by workers. The scores per task could be therefore aggregated into Mean Complexity Scores (MCS), and Mean Factor Scores (MFS) for each individual complexity factor. Figure 6.1 shows the MCS and MFS per task plotted against their related SOS, along with the curve fitting the data (p-value < 0.001 for each complexity factor) according to the SOS relationship described before.

Subjective Task Complexity Scores. We now analyse the output of the `NASA-TLX` questionnaire. We are interested in observing whether some complexity factors would get higher scores than others, and how users deemed

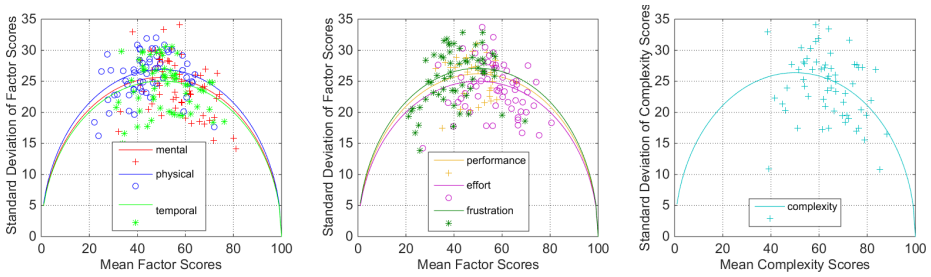


Figure 6.1: SOS Hypothesis plots for complexity (rightmost) and its factors (left and center plots). The continuous lines depict the square fitting found applying the SOS hypothesis.

each factor relevant in the final complexity score. In this respect, we computed, in addition to MCS and MFS, also Mean Factor Weights (MFW), as the average weight value given by individual evaluators (i.e., workers) to the same task.

Kolmogorov-Smirnoff and Levene tests revealed MCS, MFS and MFW to be normally and homogeneously distributed, thus allowing the application of parametric analysis. The 61 tasks obtained an average complexity value of 63.78 ± 11.56. Figure 6.2 shows the factor values scored by the 61 tasks on average. A one-way ANOVA setup with complexity factor as independent value and mean score per task as dependent variable, revealed that tasks were scored significantly different along the six factors ($F = 22.414, df = 6, p < 0.001$). Specifically, tasks scored lower (according to post-hoc tests with Bonferroni correction) in physical complexity, and highest in mental complexity and effort. Considering that online crowdsourcing tasks usually involve very little physical labour, and more mental effort, these results are in line with the expectations. Perceived performance was also scored significantly lower than mental complexity and effort ($p < 0.001$ in both cases); this indicates that workers were, on average, relatively happy with their performance with the task (the lower the score on the performance sub-scale, the better the perceived performance). Finally, frustration scored significantly lower than all other factors (except for physical complexity), which suggests workers to be rather satisfied with the tasks we instantiated.

To understand the relative importance of each factor in the overall complexity perception, we also looked at the distribution of the MFW across tasks. An ANOVA with Bonferroni correction ($F = 170.821, df = 5, p < 0.001$), established that all factors had significantly different weights for our tasks (see Figure 6.3), except for effort and mental (p = 0.767). The latter two obtained, across tasks, relatively high weights indicating that mental

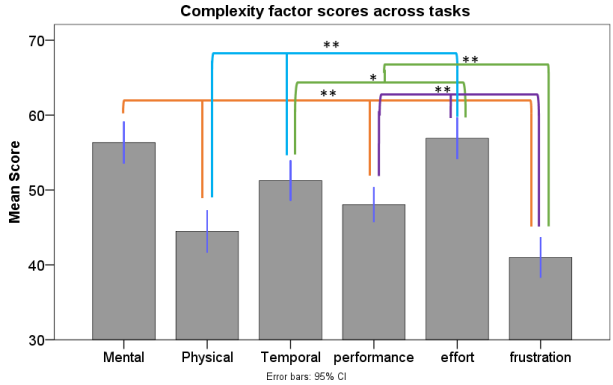


Figure 6.2: Mean Scores per complexity factors across the 61 instantiated tasks. "*" indicates mean differences with significance to the .05 level, "**" indicates mean differences with significance to the .01 level.

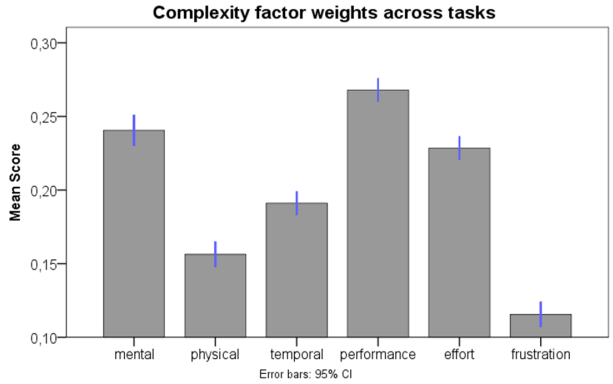


Figure 6.3: Mean values of the weights assigned to the six complexity factors throughout the 61 instantiated tasks. All means are significantly different except those of mental complexity and effort.

complexity and effort to complete the task are very relevant in judging the complexity of crowdsourcing tasks. Interestingly, the highest weight was assigned to performance. Whereas mental complexity was expected to have high relevance in the overall complexity, the fact that workers seem to care about the degree to which they perform the task properly, suggests that intrinsic motivation (as in willingness to perform the assigned job properly) may play more than a marginal role in workers satisfaction and overall motivation to complete complex crowdsourcing tasks. Frustration, which obtained on average the lowest weight, seems instead not impact majorly the overall perceived complexity of the task.

Finally, we verified whether task type (see Table 6.1) had an influence on perceived complexity, as well as task performance time, and TLX completion time. As complexity scores were normally and homogeneously distributed when clustered according to task, we used again parametric testing. Note that we excluded from the analysis the task of type "other" as we only had one datapoint for that category. Task type was found to have a significant effect on complexity ($F = 3.729$, $df = 5$, $p = 0.06$). This effect could be entirely explained by the significant difference in complexity between *Content Creation* and *Interpretation and Analysis* tasks, the latter found to be less complex than the former by an average 11.61 points ($p = 0.017$ after Bonferroni correction). Task performance time and TLX completion time were found to be non-normally distributed. A Kruskal Wallis test revealed that for both dependent variables, task type did not have a significant effect (for task execution time: $H = 9.067$, $p = 0.170$; for TLX completion time: $H = 2.159$, p = 0.866).

6.5 Task Complexity Prediction

Having collected subjective complexity scores for our 61-task dataset, we then checked whether our proposed features were useful to predict it.

Feature sets. As input for our model, we experiment separately with each of the feature categories described before (metadata, content, and visual features). We also test a fourth prediction configuration, in which a LDA is applied to content features to extract semantic topics, thereby reducing the dimensionality of the feature set for model training and testing. The dimensionality of metadata, visual and semantic features was 9, 14, and 1447, respectively.

Regression models. Due to the high dimensionality of content features, we propose a novel regression model, i.e. matrix factorization boosted linear regression (MFLR), that performs at the same time 1) dimension reduction, via non-negative matrix factorization [129], and 2) regression, via Lasso [213]. The key idea of MFLR is to learn latent complexity dimensions (LCD's) that are predictive for task complexity. To do so, the model jointly optimizes matrix factorization and regression functions as follows

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}\geq\mathbf{0},\mathbf{W}} \quad & \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_F^2+\lambda\|\mathbf{U}\mathbf{W}^T-\mathbf{Y}\|_F^2 \\ & +\lambda_X(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2)+\lambda_Y\|\mathbf{W}\|_1 \end{aligned} \tag{6.1}$$

Feature Set	Regression Model			
	Linear	Lasso	MFLR	Random Forest
Metadata	13.37±4.18	13.16±4.24	–	**9.94±1.68**
Visual	14.86±4.01	12.50±2.07	**9.97±1.28**	10.21±1.15
Content	12.87±1.64	9.97±1.27	**9.18±1.83**	10.00±1.47
Content LDA	10.34±1.84	**9.23±1.44**	–	11.80±1.18

Table 6.3: Subjective complexity estimation, measured by mean absolute error (MAE). The best predicting model for each feature class is highlighted in bold; the best performance across all features with different regression models is underlined.

where $\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|_F^2$ factorizes the task-feature matrix $\mathbf{X}$ to two smaller matrices, i.e. $\mathbf{U}$ as the task-LCD matrix and $\mathbf{V}$ as the feature-LCD matrix; $\|\mathbf{U}\mathbf{W}^T - \mathbf{Y}\|_F^2$ takes task LCD's ($\mathbf{U}$) for predicting the complexity ($\mathbf{Y}$). The regularization terms $\mathcal{L}_X(\mathbf{U}, \mathbf{V})$ and $\mathcal{L}_Y(\mathbf{W})$ are implemented with Frobenius and L_1 norm, and regulated by λ_X and λ_Y respectively [6].

In MFLR, feature importance can be computed as $F_{importance} = \mathbf{V}\mathbf{W}^T$, where feature f_i has contribution $\mathbf{V}[i]\mathbf{W}^T$, with $\mathbf{V}[i]$ denoting the distribution of feature f_i over the learned latent complexity dimensions.

To assess the performance of MFLR, we experiment with 3 methods: 1) Linear regression, used as the baseline model; 2) Lasso, which is a linear model with feature selection; and 3) Random forest, which is an ensemble model known for its good generalization capabilities.

Parameter setting. For Lasso, we use the default setting in sklearn[7], in which the regularization parameter is set to 1, which implies that the error function and the regularization term have the same weight. To make the results comparable, we also set $\lambda = \lambda_Y$ in MFLR. Using a grid search on the training data, we set the parameters of MFLR as follows: $\lambda_X = 1, \lambda = \lambda_Y = 0.01$. To account for the Content features dimensionality variance, we set the number of topics to be extracted by LDA to 10, so as the number of latent complexity dimensions of MFLR.

Results. Table 6.3 summarizes the resulting prediction performance. We ran a 5-fold cross-validation and report the averaged performance measured by mean absolute error (MAE). The ground truth task complexity is 63.78 ± 11.46 based on crowd worker annotations. The MAEs of different configurations fall into the range (9, 14), an acceptable error considering the ground truth figures. Overall, content features yield the best prediction performance

[6] We refer the reader to our companion page for the detailed optimization algorithm.
[7] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html

Visual Feature	Imp.	Semantic Feature	Imp.
visualAreaCount	3.25	linkCount	2.42
hueAvg	0.09	wordCount	1.37
		keyword: `audio`	0.09
		keyword: `transcribe`	0.07
		keyword: `writing`	0.06
imageAreaCount	-0.27	unigram: `clear`	-0.06
colorfulness1	-0.63	unigram: `identify`	-0.07
scriptCount	-1.52	uigram: `date`	-0.09
valAvg	-1.71	keyword: `easy`	-0.10
cssCount	-1.82	imageCount	-1.01

Table 6.4: Features more correlated (positively and negatively) with subjective complexity.

with `MFLR`, although LDA-reduced content features with `Lasso` follow closely. Notably, prediction with visual features is also significantly improved with `MFLR`, a result that shows the good performance achievable by `MFLR` also with a lower number of dimensions. We account these results to the fact that `MFLR` re-combines features into latent complexity dimensions, projecting the input data into a (low-dimensional) transformed space better characterizing task complexity. This has obvious benefits with respect to `Lasso`, which uses the original feature space instead; with respect to LDA, the advantage of `MFLR` is in the fact that the latent complexity factors derivation is guided by the regression, thereby increasing the predictive power of `LCD`s'.

Important features. Table 6.4 summarizes the visual and content features most positively and negatively correlated with complexity. The most positively correlated visual feature is the number of visual areas, indicating that more visual items lead to higher task complexity perceived by workers. The presence of curated layouts and interactions (CSS and Javascript) are negatively correlated features with complexity, suggesting that a better design of the task presentation and more interactive components could decrease the complexity perceived by workers. Notably, the stronger visual factor is `valAvg`, i.e. the "value" (or "lightness") of the task design. In content features, complexity is reflected from the point of view of required actions to be performed by workers (e.g., *transcribe*), task type (e.g., *writing*), and content matter (e.g. *audio*). The amount of task elements (text, images, links) also plays a relevant role.

6.6 Can Task Complexity Features Help Improve Task Performance Prediction?

As structural features of tasks have an influence on subjective complexity, we are interested in understanding if they are also helpful in predicting task performance. This section reports our efforts in applying the task complexity feature set on task performance prediction, where we approximate task performance by execution throughput. Our interest is in 1) evaluating the utility of using the complexity feature set for throughput prediction, 2) understanding the features that yield best prediction performance, and compare them with the ones best for complexity prediction.

6.6.1 Experimental Setup

Data Preparation. We operate on a five year dataset [63], collected from Amazon mTurk from 2009 to 2014. The dataset contains more than $2.56M$ distinct batches, and $130M$ HITs. All batches in the dataset are described by metadata, but only 55% of them are provided with task content. No task in the dataset contains all the information (e.g. CSS stylesheets, external resources) required to evaluate visual features. The throughput prediction problem has been previously studied in [63]. The goal is to predict the number of tasks executed in a predefined time span. We apply the same experimental methodology as in [63]. We randomly select 50 time points, uniformly sampled from 5 months (June to October 2014) of mTurk market data. For each time point, we predict the #tasks executed in the following hour, using as training data information about tasks published and executed in the previous four hours. To avoid biases due to the over-representation in the marketplace of a specific requester, we sample the task space by selecting, for each requester, a single batch per task type. This choice allows to keep an accurate distribution of task type, while avoiding compensation for the inevitable skewness introduced by big market players [63]. The result is an experimental dataset consisting of 8675 tasks. To better characterize prediction performance on batches with different throughput magnitude, we divide the dataset into three subgroups, according to the power-law distribution, with batch size varying in the range of $[1, 10)$, $[10, 100)$ and $[100, 1000)$, respectively. For evaluation, we again use the Mean Absolute Error (MAE) as a metric to measure prediction performance.

Feature Sets. As input for throughput prediction, we experiment separately with each metadata and content features. Visual features could not be tested

due to their absence in the majority of tasks. We also include task and market *dynamic* features, which are related to the execution context of the task in the marketplace. Inspired by [63], we consider properties of the marketplace in the observation interval, including: the total number of `Available HITs` in the market (to account for market size); the total number and relative size of `Published` and `Completed` HITs (to account for market speed); and the total amount of `Available` and `Obtained Rewards` (to account for market value). The resulting feature dimensionality is 9 and 11, respectively; content features dimensionality varies according the number of unigrams in the subgroup, leading to $26.4k$ ($[1, 10)$), $15.2k$ ($[10, 100)$) and $1.3k$ ($[100, 1000)$) content features, respectively.

Models and Parameter Setting. We experimented again with `Linear` regression, `Lasso` and `Random Forest` regression. In this case of task performance predict on, `MFLR` finds latent performance dimensions (`LPD`) drawn from the considered feature classes. We also applied `LDA` on content features for topic modeling. We set both the number of topics for `LDA` and the number of latent performance dimensions of `MFLR` to 100.

6.6.2 Results

Table 6.5 summarizes the prediction performance obtained by applying different feature classes to the considered regression models. The comparison of prediction performance on different feature classes highlights how dynamic features (i.e., the marketplace context) provide the least prediction support; this is an interesting result, that hints to the greater importance of objective task features for throughput prediction. We observe that metadata feature achieves good performance with `Random Forest` and content features achieves good performance with `Lasso`. This is not surprising, given that content features are of high dimensionality and `Lasso` properly select predictive features for task performance estimation. Interestingly, high-level semantic features, i.e. topics extract via LDA, achieves better performance than plain content feature with `Lasso`, which shows that dimension reduction improves the predictive power of content features. Most importantly, content features achieves the best performance with `MFLR` compared to all configurations when applied to (in batch groups $[1 - 10)$ and $[10 - 100)$), which is consistent with the complexity prediction experiment. It is worth noting that the configurations of $[1 - 10)$ and $[10 - 100)$ are the ones where the other models make the biggest errors, relatively to the range within throughput varies.

Feature Set	**[1,10) (GT: 3.01±2.35)**				**[10,100) (GT: 29.92±21.63)**				**[100,1000) (GT: 265.45±185.97)**			
	Linear	Lasso	MFLR	RForest	Linear	Lasso	MFLR	RForest	Linear	Lasso	MFLR	RForest
Metadata	3.88	3.78	–	**3.65**	18.35	18.11	–	**17.45**	126.81	126.45	–	**107.60**
Dynamic	**3.73**	3.61	–	3.93	20.31	**18.95**	–	20.47	**126.76**	131.30	–	138.55
Content	518.29	3.42	**2.72**	4.50	576.44	16.86	**15.65**	20.53	265.45	**110.14**	112.48	116.33
Content LDA	4.11	**3.42**	–	4.55	18.92	**17.88**	–	19.75	123.11	**118.15**	–	128.43

Table 6.5: Throughput prediction performance of different feature classes and regression models, measured by MAE. Results are reported for three batch groups: throughput within the range of $[1, 10)$, $[10, 100)$, and $[100, 1000)$. **GT** is the ground truth throughput. In every group, the best performance among different regression models for each class of features is highlighted in bold; the best performance among all feature classes for every group is underlined.

Impo.	$\mathbf{LPD}_1$	$\mathbf{LPD}_2$	$\mathbf{LPD}_3$	$\mathbf{LPD}_4$
LPD's	text	find	search	expressions
and	clip	company	relevance	faces
Their	copy	online	google	emotions
main	easy	fast	internet	mimicking
unigrams	quick	entry	information	camera
Impo.	bonus	copy	easy	categorization
unigrams	image	search	instances	indentification

Table 6.6: Main LPD's and unigrams that contribute the most to throughput prediction.

Content features are more informative than metadata features in low throughput groups ($[1, 10)$ and $[10, 100)$), with significant performance differences in both cases ($p < 0.01$, Wilcoxon signed-rank test); however, metadata features work better in the group $[100, 1000)$, although with non significant differences in performance ($p = 0.36$, Wilcoxon signed-rank test). These results complement previous work in throughput prediction [63], where high prediction performance is mainly guided by the `Initial Hits` metadata feature. We show how the importance of such feature significantly decreases in low throughput groups, where content features have a prominent role. These results might indicate that workers execute large batches mainly due to the high volume of HITs available to be executed, while for batches of smaller sizes the content of the task have a bigger influence on workers' decision for task execution. Therefore requesters are recommended to better design their tasks of small batch sizes.

Important Features. Table 6.6 (line 2-6) shows the latent performance dimensions of content features that contribute the most in prediction: the most important $\mathtt{LPD}_1$ contains terms related to the type of actions required from the worker (arguably, *copying* and *pasting* of *text* are a simple actions to perform); the second and third most important LPD's reflect performance from the point of view of task type (information finding – *find*, *online*, *google* – is known to be among the most demanding type of crowdsourcing tasks, in terms of worker labour). Finally, the fourth LPD hints to the type of annotation to be performed (subjective, looking for expressions and emotions), and to the type of content (images and videos). Table 6.6 (line 7-8) summarizes the most important unigrams across factors. Terms related to actions, task type, and task matter emerge as most representative.

Features for Complexity vs. Performance. Content features can be used to predict both task complexity and task performance. Are there content features able to predict both? We tested the presence of correlation

	- TP	+ TP
- CPX	know, words, exact guidelines, shown free, ask, based change, load, tell notes, number, need	photo, picture, text item, identify, type thank, best, easy address, job, pay accept, right
+ CPX	questions, answer file, provide create, options, listed save, issues, result personal, send, unable	transcribe, audio, images review, feedback, search read, article, try code, writing, sentence carefully, follow, instructions

Table 6.7: Top-15 unigrams associated with each positive **(+)** and negative **(-)** correlation with complexity (**CPX**) and throughput (**TP**) prediction.

between the *importance* of features in complexity prediction and in performance prediction. Results show a weak negative correlation of feature importance. Significance (Pearson correlation: -0.1308; $p < 0.05$) is present only for low throughput (**[1,10)**) tasks. This result supports the findings from the previous section, where low throughput tasks were the ones more likely to benefit for content features for prediction purposes. The negative correlation indicates the presence of features that, while hinting to higher task complexity, can signal low completion performance, and viceversa. We show these features in Table 6.7. The weak – yet significant – correlation is due to the existence of features having consistent correlation, i.e. both associated positively (respectively, negatively) with complexity and performance. Unigrams describing task actions and task type are again the ones more likely to be associated with consistent complexity and performance prediction. For instance *transcribe audio* unigrams are predictive of high complexity[8] and higher throughput in the low throughput (**[1,10)**) batches. This is an unexpected result, that we can explain only by looking at the mTurk market as a whole [63], where the presence of large batches devoted to audio transcription might influence workers' task selection strategy. Clearly, further investigation focusing on the relationship between task complexity, market dynamics, and execution performance is needed, and it will be tackled in future work.

6.7 Conclusion

This work studied the subjective complexity (i.e., as perceived by workers) of crowdsourcing tasks. We defined (1) an operational way to quantify subjective complexity; (2) a set of objective features (i.e., quantities computable

[8] Arguably, audio transcription requires a lot of attention for proper execution.

from the task metadata and HTML code) from which to predict complexity automatically; and (3) a novel regression model, `MFLR`, that shows superior performance for complexity prediction. We were able to deliver a model predicting subjective task complexity in an accurate and fully automatic way; features describing the semantic content of the task are most predictive for complexity, followed closely by the features describing the visual appearance of the task. We show how this feature set is also useful to improve the prediction of task throughput when workers' task selection strategy is not influenced by batch size.

Our automatic complexity predictors have the potential to impact crowdsourcing research widely. We expect them to be useful in deploying new strategies for workers retention by, e.g., adjusting task complexity over a batch of tasks. We also expect that measuring task complexity will create a better communication channel between requesters and workers thanks to the shared understanding of the required effort to complete the task, as well as the implementation of more fair compensation mechanisms.

Chapter 7

On the Role of Task Clarity in Microtask Crowdsourcing

This chapter studies another important task property, namely, task clarity. Through a survey with 100 workers in human computation systems, we show that unclear tasks is indeed a big concern for workers in task execution. We then introduce a method for task clarity modeling based on the *goal* and *role* clarity constructs, and a novel method for measuring task clarity. We further show that task clarity is highly uncorrelated with task complexity, thus contributing a new understanding of the relationship between these two important properties of tasks in knowledge creation.

This chapter is published as "Clarity is a Worthwhile Quality - On the Role of Task Clarity in Microtask Crowdsourcing" [72], by U. Gadiraju, J. Yang, and A. Bozzon in Proceedings of the 28th ACM conference on Hypertext and Social Media. ACM, 2017.

7.1 Introduction

Microtask crowdsourcing has become an appealing approach for data collection and augmentation purposes, as demonstrated by the consistent growth of crowdsourcing marketplaces such as Amazon Mechanical Turk[2] and CrowdFlower[3].

Task consumption in microtask crowdsourcing platforms is mostly driven by a self-selection process, where workers meeting the required eligibility criteria select the tasks that they prefer to work on. Workers strive to maintain high reputation and performance to access more tasks, while maximizing monetary income. When discussing such a trade-off, the dominant narrative suggests that workers are more interested in obtaining their rewards, than in executing good work. We challenge this widespread opinion by focusing on an often neglected component of microtask crowdsourcing: the *clarity* of task description and instructions in terms of comprehensibility for workers.

Poor formulation of tasks has clear consequences: to compensate for a lack of alternatives in the marketplace, workers often attempt the execution of tasks despite a sub-optimal understanding of the work to be done. On the other hand, requesters are often not aware of issues with their task design, thus considering unsatisfactory work as evidence of malicious behaviour and deny rewards. As a result, crowd workers get demotivated, the overall quality of work produced decreases, and all actors lose confidence in the marketplace. Despite the intuitive importance of task *clarity* for microtask crowdsourcing, there is no clear understanding of the extent by which the lack of clarity in task description and instructions impacts worker performance, ultimately affecting the quality of work.

Research Questions and Original Contributions. This chapter aims at filling this knowledge gap by contributing novel insights on the nature and importance of task clarity in microtask crowdsourcing. By combining qualitative and quantitative analysis, we seek to answer the following research questions.

- **RQ1**: What makes the specification of a task unclear to crowd workers? How do workers deal with unclear tasks?

First, we investigate if clarity is indeed a concern for workers. We designed and deployed a survey on the CrowdFlower platform, where we asked workers

[2]http://www/mturk.com/
[3]http://www.crowdflower.com/

to describe what makes a task unclear, and to illustrate their strategies for dealing with unclear tasks. The survey involved 100 workers, and clearly highlights that workers confront unclear tasks on a regular basis.

Some workers attempt to overcome the difficulties they face with inadequate instructions and unclear language by using external help, dictionaries or translators. Several workers tend to complete unclear tasks despite not understanding the objectives entirely.

These results demonstrate the need for methods for task clarity measurement and prediction, and shaped the formulation of the following questions.

- **RQ2**: How is the clarity of crowdsourcing tasks perceived by workers, and distributed over tasks?

Inspired by work performed in the field of organizational psychology, we consider clarity both in the context of *what* needs to be produced by the worker (*goal clarity*) and *how* such work should be performed (*role clarity*). We sampled 7.1K tasks from a 5 years worth dataset of the Amazon mTurk marketplace. Tasks were published on CrowdFlower to collect clarity assessments from workers. Results show that task clarity is coherently perceived by crowd workers, and is affected by the type of the task. We unveil a significant lack of correlation between the *clarity* and the *complexity* of tasks, thus showing that these two properties orthogonally characterize microwork tasks.

- **RQ3**: Which features can characterize the goal and role clarity of a task? Using such features, to what extent can task clarity be predicted?

We propose a set of features based on the metadata of tasks, task type, task content, and task readability to capture task clarity. We use the acquired labels to train and validate a supervised machine learning model for task clarity prediction. Our proposed model to predict task clarity on a 5-point scale achieves a mean absolute error (*MAE*) of *0.4 (SD=.003)*, indicating that task clarity can be accurately predicted.

- **RQ4**: To what extent is task clarity a macro-property of the Amazon mTurk ecosystem?

We analyzed 7.1K tasks to understand how task clarity evolves over time. We found that the overall task clarity in the marketplace fluctuates over time,

albeit without a discernible pattern. We found a weak positive correlation between the average task clarity and the number of tasks deployed by requesters over time, but no significant effect of the number of tasks deployed by requesters on the magnitude of change in task clarity.

7.2 Related Literature

Text readability. Readability has been defined as the sum of all elements in text that affect a reader's understanding, reading speed and level of interest in the material [55]. There has been a lot of work in the past on analyzing the readability of text, as summarized in [49]. Early works range from simple approaches that focus on the semantic and syntactic complexity of text [120], or vocabulary based approaches where semantic difficulty is operationalized by means of gathering information on the average vocabulary of a certain age or social status group [39]. More recently, authors proposed statistical language models to compute readability [50]. Other works studied the lexical richness of text by capturing the range and diversity of vocabulary in given text [145]. Several machine learning models have also been proposed to predict the readability of text [173, 117]. De Clerq et al. recently investigated the use of crowdsourcing for assessing readability [56]. The vast body of literature corresponding to text readability has also resulted in several software packages and tools to compute readability [80, 53].

In this chapter, we draw inspiration from related literature on text readability in order to construct features that aid in the prediction of task clarity on crowdsourcing platforms.

Task Clarity in Microtask Crowdsourcing. Research works in the field of microtask crowdsourcing have referred to the importance of task clarity tangentially; several authors have stressed about the positive impact of task design, clear instructions and descriptions on the quality of crowdsourced work [147, 195, 121, 18]. Grady and Lease pointed out the importance of wording and terminology in designing crowdsourcing tasks effectively [79]. Alonso and Baeza-Yates recommended providing 'clear and colloquial' instructions as an important part of task design [7]. Kittur et al. identified 'improving task design through better communication' as one of the pivotal next steps in designing efficient crowdsourcing solutions in the future [122]. The authors elaborated that task instructions are often ambiguous and incomplete, do not address boundary cases, and do not provide adequate examples. Khanna et al. studied usability barriers that were prevalent on

Amazon mTurk (AMT), which prevented workers with little digital literacy skills from participating and completing work on AMT [118]. The authors showed that the task instructions, user interface, and the workers' cultural context corresponded to key usability barriers. To overcome such usability obstacles on AMT and better enable access and participation of low-income workers in India, the authors proposed the use of simplified user interfaces, simplified task instructions, and language localization. More recently, Yang et al. investigated the role of *task complexity* in worker performance, with an aim to better the understanding of task-related elements that aid or deter crowd work [232].

While the importance of task clarity has been acknowledged in the microtask crowdsourcing community, there is neither a model that describes task clarity nor a measure to quantify it. In this chapter, we not only propose a model for task clarity, but we also present a means to measure it. To the best of our knowledge, this is the first work that thoroughly investigates the features that determine task clarity in microtask crowdsourcing, and provides an analysis of the evolution of task clarity.

Task Clarity in Other Domains. In the field of organizational psychology, researchers have studied how the sexual composition of groups affects the authority behavior of group leaders in cases where the task clarity is either *high* or *low* [189]. In this case, the authors defined task clarity as the degree to which the goal (i.e., the desired outcome of an activity) and the role (i.e., the activities performed by an actor during the course of a task) are clear to a group leader. In self-regulated learning, researchers have widely studied task interpretation as summarized in [184]. Hadwin proposed a model that suggests the role of the following three aspects in task interpretation and understanding; (i) implicit aspects, (ii) explicit aspects, and socio-contextual aspects [87, 86]. Recent literature regarding task interpretation in the learning field has revolved around text decoding, instructional practices or perceptions of tasks on the one hand [33, 107, 141], and socio-contextual aspects of task interpretation such as beliefs about expertise, ability, and knowledge on the other hand [37, 54].

Inspired by the modeling of task clarity in the context of authority behavior in Psychology, we model task clarity as a combination of *goal clarity* and *role clarity* (as explained in Section 7.4).

7.3 Are Crowdsoucred Microtasks always clear?

We aim to investigate whether or not workers believe task clarity to impact their work performance (**RQ1**). We thereby deployed a survey consisting of various questions ranging from general demographics of the crowd to questions regarding their experiences while completing microtasks on crowdsourcing platforms.

7.3.1 Methodology

We deployed the survey on CrowdFlower[4] and gathered responses from 100 distinct crowd workers. To detect untrustworthy workers and ensure reliability of the responses received, we follow recommended guidelines for ensuring high quality results in surveys [71]. To this end, we intersperse two attention check questions within the survey. In addition, we use the filter provided by CrowdFlower to ensure the participation of only high quality workers (i.e., *level 3* crowd workers as prescribed on the CrowdFlower platform). We flagged workers who failed to pass at least one of the two attention check questions and do not consider them in our analysis.

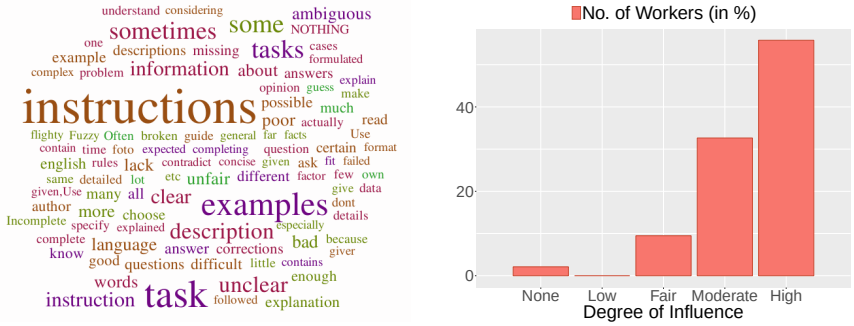
7.3.2 Analysis and Findings

Worker's Experience. We found that around 36% of the workers who completed the survey earn their primary source of income through crowd work. 32.6% of the workers claim to have been contributing piecework through crowdsourcing platforms over the last 3 to 6 months. 63.2% of the workers have been doing so for the last 1 to 3 years. A small fraction of workers (3.2%) claim to have been working on microtasks for the last 3 to 5 years, while 1% of the worker population has been contributing to crowdsourced microtasks for over 5 years. During the course of this time, almost 74% of workers claim to have completed over 500 different tasks.

What factors make tasks unclear? We asked the workers to provide details regarding the factors that they believe make tasks unclear, in an open text field. The word-cloud in Figure 7.1(a) represents the responses collected from the crowd workers. Workers complained about the task instructions and descriptions being '*vague*', '*blank*', '*unclear*', '*inconsistent*', '*imprecise*', '*ambiguous*', or '*poor*'. Workers also complained about the language used; '*too many words*', '*high standard of English*', '*broken English*', '*spelling*', and so

[4] http://crowdflower.com

forth. Workers also pointed out that adequate examples are seldom provided by requesters. Excerpts of these responses are presented on the companion webpage[5].



(a) (b)

Figure 7.1: (a) Word-cloud representing factors cited by workers that make tasks unclear. Size of words indicate frequency. (b) Degree of influence of task clarity on performance.

Task Clarity and Influence on Performance. Around 49% of workers claimed that up to a maximum of 30% of the tasks that they worked on were unclear. 37% of workers claimed that between 31-60% of the tasks they completed lacked clarity, while 14% of the workers claimed that more than 60% of their completed tasks were unclear. We also asked the workers about the perceived influence of task clarity on their performance. Our findings are presented in the Figure 7.1(b). A large majority of workers believe that task clarity has a quantifiable influence on their performance. We also asked workers about the frequency of encounter for tasks containing difficult words, which might have hindered their performance. Figure 7.2(a) depicts our findings, indicating that workers observed tasks which contained difficult words reasonably frequently.

How do workers deal with unclear tasks? We investigated the frequency with which workers complete tasks despite the lack of clarity. As shown in Figure 7.2(b), we found that nearly 27% of workers complete less than 10% of the unclear tasks that they encounter.

On the other hand, another 27% of workers completed more than 50% of all the unclear tasks they come across. In addition, around 18% of workers used dictionaries or other helpful means/tools to better understand over 50%

[5]https://sites.google.com/site/ht2017clarity/

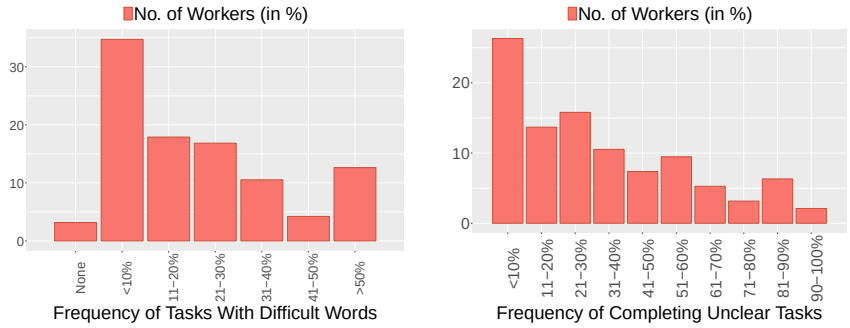


Figure 7.2: (a) Frequency of tasks with difficult words, and (b) frequency of workers completing unclear tasks.

of tasks they completed. 20% of workers used translators in more than 50% of the tasks that they completed.

7.4 Modeling Task Clarity

We address **RQ2** by modeling task clarity of crowdsourced microtasks as a combination of *goal clarity* and *role clarity*. Inspired by previous work in organizational psychology [189], we define task clarity as a combination of the extent to which the desired outcome of a task is clear (goal clarity), and the extent to which the workflow or activities to be carried out are clear (role clarity).

7.4.1 Assessing Task Clarity

Task clarity of microtasks in a marketplace is a notion that can be quantified by human assessors by examining task metadata such as the *title*, *keywords* associated with the task, *instructions* and *description*. Since these are the main attributes that requesters use to communicate the desired outcomes of the tasks, and prescribe how crowd workers should proceed in order to realize the objectives, we argue that they play an important role in shaping crowd work.

7.4.2 Acquiring Task Clarity Labels

With an aim to understand the distribution of task clarity across the diverse landscape of tasks on AMT [63], we sampled 7,100 tasks that were deployed on AMT over a period of 1 year between October 2013 to September 2014. For every month spanning the year, we randomly sampled 100 tasks of each of the 6 task types proposed in previous work [70]; *content creation* (CC), *information finding* (IF), *interpretation and analysis* (IA), *verification and validation* (VV), *content access* (CA)[6] and *surveys* (SU). Next, we deployed a job[7] on CrowdFlower to acquire task clarity labels from crowd workers. We first provided detailed instructions describing task clarity, goal clarity and role clarity. An excerpt from the task overview is presented below:

"Task clarity defines the quality of a task in terms of its comprehensibility. It is a combination of two aspects; (i) goal clarity, i.e, the extent to which the objective of the task is clear, and (ii) role clarity, i.e., the extent to which the steps or activities to be carried out in the task are clear."

In each task workers were required to answer 10 questions on a 5-point Likert scale. The questions involved assessing the goal and role clarity of the corresponding task, the overall task clarity, the influence of goal and role clarity in assessing overall task clarity, clarity of title, instructions and description, the extent to which the title conveyed the task description, the extent to which the keywords conveyed the task description and goal of the task, and the quality of language in the task description. Apart from these 10 questions, workers were provided with an optional text field where they could enter comments or remarks about the AMT task they evaluated. We gathered responses to these questions for each of the 7,100 tasks from 5 distinct crowd workers. We controlled for quality by using the *highest quality* restriction on CrowdFlower, that allows only workers with a near perfect accuracy over hundreds of different tasks and varying task types. In addition, we interspersed attention check questions where workers were asked to enter alphanumeric codes that were displayed to them. In return, workers were compensated according to the hourly rate of 7.5 USD.

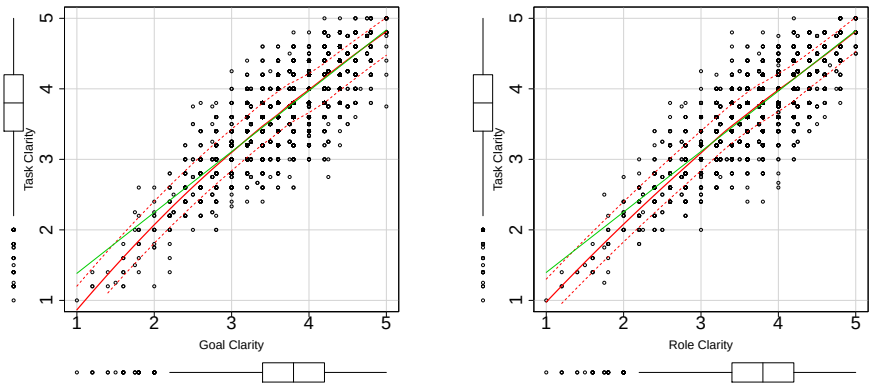
[6]Note that there were fewer than 100 tasks of the CA type in a few months during the time period considered. In those cases, we considered all available tasks.

[7]Preview available in the companion page: https://sites.google.com/site/ht2017clarity/.

7.4.3 Perception of Task Clarity

We found that the mean task, goal and role clarity across the different tasks were nearly the same. On average, workers perceived tasks to be moderately clear (*M=3.77, SD=.53*). The same is the case with goal clarity (*M=3.76, SD=.53*) and role clarity (*M=3.76, SD=0.54*). On investigating the influence of goal and role clarity on the crowd workers in adjudicating the overall task clarity, we found that role clarity and goal clarity were both important in determining task clarity. On average, workers responded that goal clarity influenced their judgment of overall task clarity to an extent of 3.98/5 (*SD=.51*), and that in case of role clarity was 3.93/5 (*SD=.52*). We found that goal clarity was slightly more influential than role clarity in determining the task clarity, and this difference was statistically significant; $t(14199) = 25.28, p < .001$.

We also analyzed the relationship of task clarity with goal and role clarity respectively. We found strong positive linear relationships in both cases, as shown in Figure 7.3.



(a) (b)

Figure 7.3: Relationship of Task Clarity with (a) Goal Clarity, and (b) Role Clarity. The trendline is represented in green, and the regression line is represented by the thick red line.

We computed Pearson's **r** between task clarity with each of goal and role clarity; $r(14998) = .85, R^2 = .72, p < .001$ and $r(14998) = .86, R^2 = .74, p < .001$. These findings indicate that it is equally important for task requesters to ensure that the objective of the task, as well as the means to achieve the desired outcome are adequately communicated to crowd workers via the task title, instructions and description, and keywords associated with the task.

Inter-worker Agreement. To find out whether or not task clarity is coherently perceived by workers, we verify the presence of agreement of task clarity evaluations among workers. Given the subjective nature of task clarity evaluations, we apply the SOS Hypothesis [98], which examines the extent to which individual evaluations of clarity spread around the mean clarity value per task. The SOS Hypothesis has proven to be more reliable than other inter-evaluator agreement measures such as Krippendorff's alpha, in subjective assessment tasks that involve a set of participants evaluating the same item – in our case, the same task [8]. In SOS Hypothesis, we test the magnitude of the squared relationship between the standard deviation (i.e. SOS) of the evaluations and the mean opinion score (MOS; in our case, mean clarity score), denoted by α. The value of α can then be compared with those of other subjective assessment tasks that are deemed to be more (high α) or less prone to disagreement (low α) among evaluators. Specifically, for 5-point scale evaluations, SOS Hypothesis tests a square relationship between SOS and MOS by fitting the following equation:

$$SOS(i) = -\alpha MOS(i)^2 + 6\alpha MOS(i) - 5\alpha$$

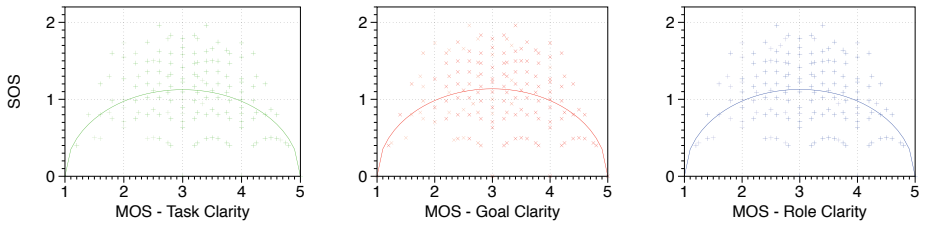
considering each task i in the evaluation pool.

Clarity	Task Clarity	Goal Clarity	Role Clarity
α	0.3166	0.3229	0.3184

Table 7.1: SOS Hypothesis α values for Task Clarity, Goal Clarity and Role Clarity.

Table 7.1 shows the α values computed for task clarity, goal clarity and role clarity. All these evaluations have a value of 0.32, which is similar to what could be obtained in other subjective assessment tasks such as smoothness of web surfing, VoIP quality, and cloud gaming quality [98]. We therefore consider it acceptable. Figure 7.4 shows the fitted quadratic curve against worker evaluations for individual tasks. A significant correlation could be obtained between the fitted SOS value and the actual SOS value (Pearson's $r = 0.506$, $p < .001$). In conclusion, we find that task clarity is coherently perceived by workers. The substantial evidence of workers' agreement in perceiving task clarity helps establish the mean clarity score as ground truth for modeling task clarity using objective task features, as we report in Section 7.5.

Task Types and Perception of Task Clarity.



(a) (b) (c)

Figure 7.4: SOS Hypothesis plots for Task Clarity (green), Goal Clarity (red), and Role Clarity (blue). The quadratic curve depicts the fitting to worker evaluations for individual tasks.

We investigated the impact of task types on the perception of task clarity and the constructs of goal and role clarity. We note that Levene's test for homogeneity of variances was not violated across the different task types with respect to each of task, goal and role clarity. We conducted a one-way between workers ANOVAs to compare the effect of task types on the perception of task, goal and role clarity respectively. We found a significant effect of task type on the perception of task clarity at the $p < .01$ level, across the 6 task type conditions; $F(5, 7002) = 6.176, p < .001$. Post-hoc comparisons using the Tukey HSD test indicated that the perception of task clarity in some task types was significantly poorer than others; as presented in Table 7.2.

Task Type	M	SD	Comparison	Tukey HSD p-value
CA	3.75	.51	*CA vs SU*	0.011*
CC	3.76	.51	*CA vs VV*	0.004**
IA	3.74	.52	*CC vs SU*	0.046*
IF	3.77	.52	*CC vs VV*	0.020*
SU	3.82	.50	*IA vs SU*	0.001**
VV	3.82	.48	*IA vs VV*	0.001**

Table 7.2: Post-hoc comparisons using the Tukey HSD test to investigate the effect of task types on *task clarity*. Comparisons resulting in significant outcomes are presented here.
(***** indicates $p < .05$ and ****** indicates $p < .01$)

We also found a significant effect of task type on (i) the perception of goal clarity at the $p < .01$ level, across the 6 task type conditions; $F(5, 7002) = 5.918, p < .001$, and (ii) the perception of role clarity at the $p < .01$ level, across the 6 task type conditions; $F(5, 7002) = 8.074, p < .001$. Post-hoc comparisons using the Tukey HSD test (Tables 7.3 and 7.4) indicated that the perception of goal and role clarity in some task types was significantly poorer than others.

Task Type	M	SD	Comparison	Tukey HSD p-value
CA	3.76	0.52	*CA vs VV*	0.006**
CC	3.76	0.50	*CC vs VV*	0.004**
IA	3.74	0.51	*IA vs SU*	0.005**
IF	3.78	0.52	*IA vs VV*	0.001**
SU	3.82	0.51		
VV	3.83	0.50		

Table 7.3: Post-hoc comparisons using the Tukey HSD test to investigate the effect of task types on *goal clarity*. Comparisons resulting in significant outcomes are presented here.
(***** indicates $p < .05$ and ****** indicates $p < .01$)

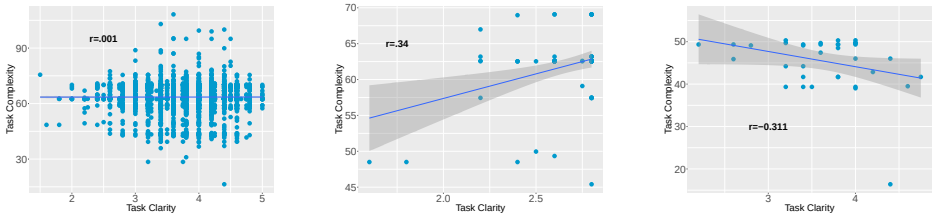
Task Type	M	SD	Comparison	Tukey HSD p-value
CA	3.75	0.51	*CA vs SU*	0.030*
CC	3.76	0.50	*CA vs VV*	0.001**
IA	3.73	0.52	*CC vs SU*	0.048*
IF	3.78	0.51	*CC vs VV*	0.001**
SU	3.82	0.50	*IA vs SU*	0.001**
VV	3.84	0.48	*IA vs VV*	0.001**
			IF vs VV	0.043*

Table 7.4: Post-hoc comparisons using the Tukey HSD test to investigate the effect of task types on *role clarity*. Comparisons resulting in significant outcomes are presented here.
(***** indicates $p < .05$ and ****** indicates $p < .01$)

7.4.4 Task Clarity and Task Complexity

Recent work by Yang et al. modeled *task complexity* in crowdsourcing microtasks [232]. By using the task complexity predictor proposed by the authors, we explored the relationship between task clarity and task complexity. We found no significant correlation between the two variables across the different types of 7,100 tasks in our dataset (see Figure 7.5(a)). This absence of a linear relationship between task complexity and task clarity suggests that tasks with high clarity can still be highly complex or tasks with low clarity can have low task complexity at the same time.

We analyzed the relationship between *task clarity* and *complexity* across different types of tasks, and found that there is no observable correlation between the two variables across the different types of tasks. As can be observed from Figure 7.3, a majority of tasks are perceived to lie within the range of moderate to high clarity. We therefore further investigated tasks with low clarity or complexity.



(a) Overall relationship between *task clarity* and *task complexity*.

(b) Relationship in information finding (IF) tasks with low *task clarity*.

(c) Relationship in content access (CA) tasks with low *task complexity*.

Figure 7.5: Relationship between *task clarity* and *complexity*.

Relationship in Tasks with Low Clarity.

As shown earlier, task clarity was coherently perceived by workers. We reason that tasks corresponding to a clarity rating < 3 have relatively *low* clarity. We investigated the effect of task types on the relationship between task clarity and complexity in tasks with low clarity. Using Pearson's **r**, we found a weak positive linear relationship between the two variables in information finding (IF) tasks with low clarity (see Figure 7.5(b)); N=*80*, **r**=*.34*. This can be explained as a consequence of complex workflows required to complete some IF tasks, where high task complexity is concomitant with relatively high task clarity. Accordingly, in IF tasks with low clarity, task complexity accounted for 11.56% of the variance in task clarity (the coefficient of determination, R^2=*.1156, p<.01*). We did not find a significant relationship between the two variables in the low clarity subsets of other task types.

Relationship in Tasks with Low Complexity.

Similarly, we consider tasks having a complexity score < 50 have relatively *low* complexity. We investigated the effect of task types on the relationship between task clarity and complexity in tasks with low complexity. Using Pearson's **r**, we found a weak negative linear relationship between the two variables in content access (CA) tasks with low complexity (see Figure 7.5(c)); N=*41*, **r**=*.311*. Thus, in CA tasks with low complexity, task clarity accounted for 9.67% of the variance in task complexity (the coefficient of determination, R^2=*.0967, p<.05*).

Discussion. The lack of linear correlation between clarity and complexity yields interesting observations. While surprising (intuitively, one might assume that a better task formulation – high clarity – would yield a lower complexity), this result is aligned with the classical theory on cognitive load,

by Sweller and Chandler [209]. The theory postulates the presence of two sources of cognitive load: *intrinsic* and *extraneous*. Intrinsic cognitive load refers to the inherent difficulty in the content of presented material, which approximates task complexity in our context; extraneous cognitive load, on the other hand, refers to the organization and presentation of material, i.e. task clarity in our context. Sweller and Chandler suggest in their theory that, while the intrinsic cognitive load is unalterable, the extraneous cognitive load can either be increased because of inappropriate instructional design, or be reduced by well-structured presentation. We show that the theory can find application in microtask crowdsourcing, as tasks of similar complexity can either be of high clarity or low clarity.

When considering tasks of specific types, however, we find correlation could be established. Specifically, we find a negative correlation with content access (CA) tasks, thus suggesting that (poorly formulated) tasks asking workers to interact with on line content (e.g. watch a video, click a link) can be perceived more complex to execute. With information finding (IF) tasks, high task complexity maps with high clarity, thus suggesting that requests for complex finding and retrieval operations can be associated with clearer instructions. These results provide further insights into the relationship between task clarity and complexity, and call for further investigation.

7.5 Prediction of Task Clarity

In this section we tackle **RQ3** and propose to model task clarity based on objective features that are extractable from tasks. We envision a system that could automatically predict task clarity and thus provides feedback to requesters on task design and to workers on task selection and execution. To test the feasibility of this idea, our study starts by designing task features that are potentially predictive for task clarity; we then build a predictive model to automatically learn task clarity based on these features.

7.5.1 Features Sets

We explore four classes of task features, namely: *metadata features*, *task type features*, *content features*, and *readability features*. In the following we provide a brief introduction to each feature class, and refer the readers to the companion page for a full description of the feature set.

Metadata Features are the task attributes associated with the definition of tasks when they are created. Typical metadata features include the number of `initial HITs`, attributes of the descriptions about desired activities to be performed by workers (e.g., `title length` and `description length`), the required qualification of workers (e.g., worker `location` and minimum `approval rate`), the estimated execution time (i.e. `allotted time`) and `reward`. These features characterize a task from different aspects that might be correlated with task clarity. For example, we assume that a longer description could entail more efforts from the requester in explaining the task.

Task Type Features categorize a task into one of the six task types defined by [70]. They are therefore high level features that comprehensively describe what knowledge is in demand. Through previous analysis, we have observed that task type has a significant effect on the perception of task clarity. We therefore assume that task type could be indicative of task clarity in prediction.

Content Features capture the semantics of a task. These features use the high-dimensional bag of words (BOW) representation. To maximize the informativeness of the content features while minimizing the amount of noise, one-hot (i.e. binary) coding was applied to the BOW feature of task title and keywords, while TF-IDF weighting was applied to the BOW feature of task description. It has been shown by research in related domains (e.g., community Q&A systems [228]) that the use of words is indicative of the quality of task formulation, therefore we are interested in understanding the effect of language use on workers' perception of task clarity.

Readability Features are by nature correlated with task clarity: tasks with higher readability are better formulated, and are thereby expected to have a higher clarity. We experiment with several widely used readability metrics in our clarity prediction task to understand their predictive power of task clarity. These include the use of long words (`long_words`), long sentence (`words_per_sentence`), the use of `preposition`, `nominalization`, and more comprehensive readability metrics such as `ARI`, `LIX`, and in particular, `Coleman_Liau`, which approximates the U.S. grade level necessary to comprehend a piece of text.

7.5.2 Prediction Results

Due to the high dimension of the content features (size of vocabulary = 10,879), we apply the Lasso method, which does feature selection and re-

gression simultaneously. We adopt 5-fold cross-validation and mean absolute error (MAE) for evaluation. Table 7.5 shows the prediction results. The prediction on task clarity achieves a MAE of 0.4032 ($SD = 0.0031$). The relatively small error compared to the scale of ground truth (i.e. 1-5) indicates that task clarity can be predicted accurately. In addition, the small standard deviation shows that the prediction is robust across different tasks. Similar results also hold for the prediction of goal clarity and role clarity, which confirms our previous observation that both are highly correlated with the overall task clarity.

Clarity	Task Clarity	Goal Clarity	Role Clarity
MAE	0.4032±0.0031	0.4076±0.0067	0.4008± 0.0070

Table 7.5: Prediction results for Task Clarity, Goal Clarity and Role Clarity, shown by $\mu \pm \sigma$.

Predictive Features. In the following we analyze the predictive features selected by Lasso. Table 7.6 shows the features with positive and negative coefficients in the Lasso model after training for task clarity prediction, i.e. features that are positively and negatively correlated with task clarity. Similar observations can be obtained for predicting goal and role clarity.

With regard to metadata features, it can be observed that longer descriptions and more keywords are positively correlated with task clarity. This suggests that more description and keywords could potentially improve the clarity of task formulation. We also observe that the increased use of images, or less use of external links could enhance task clarity. These are reasonable, since intuitively, images can help illustrate task requirements, while external links would bring in extra ambiguity to task specification in the absence of detailed explanations.

With regard to task type features, we find that tasks of type SU and VV are in general of higher clarity, while tasks of type IA are of lower clarity. This result confirms our previous findings.

With regard to content features, we observe that keyword features are more predictive than other types of content features (e.g. words in title or description). Predictive keywords include `audio`, `transcription`, `survey`, etc., which can actually characterize the majority of tasks in AMT. We therefore reason that workers' familiarity with similar tasks could enhance their perception of task clarity.

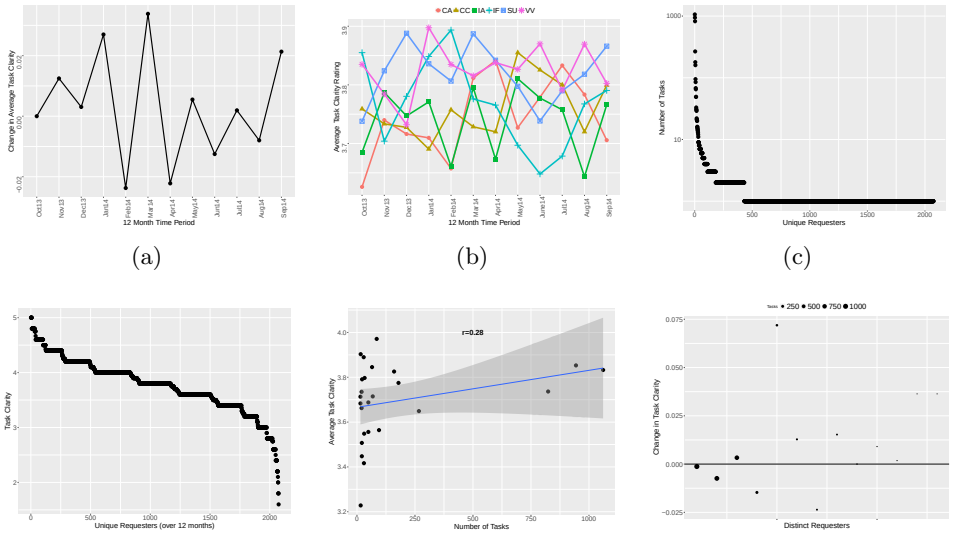
Feat. Class	Feat. w. Positive Coef.		Feat. w. Negative Coef.	
	Feature	Coef.*	Feature	Coef.*
Metadata	number_keywords	0.719	external_links	-0.598
	description_length	0.295		
	number_images	0.071		
	total_approved	0.011		
Task Type	VV	0.434	IA	-0.922
	SU	0.413		
Content	keyword: audio	2.673	keyword: id	-2.658
	keyword: transcription	1.548		
	keyword: survey	1.178		
Readability	preposition	1.748	ARI	-1.982
	GunningFogIndex	1.467	long_words	-0.671
	Coleman_Liau	0.855	syllables	-0.478
	words_per_sentence	0.620	nominalization	-0.136
	characters	0.237	pronoun	-0.104
	LIX	0.150	FleschReadingEase	-0.075
			RIX	-0.038
	(all about title)		(all about title)	

* For the sake of comparison, each value is shown with original coefficient $\times 10^2$.

Table 7.6: Predictive features for task clarity prediction.

Finally, several interesting findings with regard to task readability are observed as follows. First, many types of readability scores are indicative of task clarity, indicating a strong correlation between task readability and task clarity. Second, compared with description or keyword readability, title readability is most predictive of task clarity. As an implication for requesters, putting efforts in designing better titles can improve task clarity. Third, we observe a positive correlation between task clarity and `Coleman_Liau`, which approximates the U.S. grade level necessary to comprehend the text. The increase of Coleman_Liau (i.e. more requirements on workers' capability to comprehend the title) therefore does not lead to lower task clarity perceived by workers. The result is not surprising, given the demographic statistics of crowdworkers [63]. However, it raises questions on the suitability of this class of microtask crowdsourcing tasks for other types of working population.

On decomposing `Coleman_Liau` and exploring the effect of length of words (in terms of #letters) and length of sentences (in terms of #words), it can be observed that longer words (i.e., `long_words`) would decrease task clarity, while longer sentence (i.e., `words_per_sentence`) can enhance task clarity. This suggests that workers can generally comprehend long sentences, while the use of long words would decrease task clarity. This is consistent with



(a) (b) (c)

(d) (e) (f) # Answers

Figure 7.6: (a) Evolution of overall task clarity and (b) with respect to different types of tasks from Oct'13-Sep'14, (c) distribution of tasks corresponding to requesters who deployed them, (d) distribution of the average task clarity of tasks corresponding to distinct requesters across the 12 months, (e) relationship between the average task clarity and the number of tasks deployed by experienced requesters, (f) $\Delta TaskClarity$ of requesters who deployed tasks during more than 6/12 months in our dataset.

our findings from **RQ1**, where workers identified difficult words as a factor that decreased task clarity and also suggested that tasks with difficult words are commonplace in the microtask crowdsourcing market. We also found a positive correlation between `preposition` and task clarity, in contrast to the negative correlation between `syllables` (or `nominalization`) and task clarity. These results suggest that partitioning sentences with prepositions could increase task clarity, while complicating individual words decreases task clarity.

7.6 Evolution of Task Clarity

7.6.1 Role of Task Types

To address **RQ4**, we investigated the evolution of task clarity over time (see Figure 7.6). We found that there was no monotonous trend in the overall average task clarity over time, as shown in Figure 7.6(a). We also investigated

the effect of task type on the evolution of task clarity. We found no discernible trend in the evolution of task clarity of different types of tasks over the 12 month period considered in the dataset (Figure 7.6(b)). We conducted a one-way ANOVA to compare the effect of task type on the evolution of task clarity over time. We did not find a significant effect of task type on the evolution of task clarity at the $p < .05$ level, across the 6 task type conditions; $F(5, 66) = 0.081, p = .994$.

These findings suggest that the overall task clarity in the marketplace varies over time but does not follow a clear pattern. This can be attributed to the organic influx of new task requesters every month [63]. To identify whether the experience of task requesters plays a role in the evolution of task clarity, i.e., whether individual requesters deploy tasks with increasing task clarity over time we investigated the role of requesters in the evolution of task clarity.

7.6.2 Role of Requesters

Recent analysis of the AMT marketplace, revealed that there is an organic growth in the number of active requesters and a constant growth in the number of new requesters (at the rate of 1,000 new requesters per month) on the platform [63]. Poor task design leading to a lack of task clarity can be attributed to inexperienced requesters. To assess the role of requesters in the evolution of task clarity, we analyzed the evolution of task clarity of different types of tasks with respect to individual requesters.

We analyzed the distribution of unique requesters corresponding to the 7.1K tasks in our dataset. We found that a few requesters deployed a large portion of tasks, as depicted by the power law relationship in Figure 7.6(c). We also found that over 40% of the requesters exhibited an overall average task clarity of $\geq 4/5$, and in case of nearly 75% of the requesters it was found to be over 3.5/5 (as presented in Figure 7.6(d)). We considered requesters who deployed ≥ 15 tasks within the 12-month period as being experienced requesters, and analyzed the relationship between the number of tasks they deployed with the corresponding overall task clarity. Using Pearson's **r**, we found a weak positive correlation between the average task clarity and the number of tasks deployed by experienced requesters (see Figure 7.6(e)); **r**= .28. Thus, the experience of requesters (i.e., the number of tasks deployed) explains over 8% of the variance in the average task clarity of tasks deployed by the corresponding requesters; the coefficient of determination, $R^2 = 0.081$.

Considering the requesters who deployed tasks during more than 6 months in the 12-month period, we investigated the overall change in terms of average task clarity of the tasks deployed from one month to the next. We measure the overall change in task clarity for each requester using the following equation.

$$\Delta TaskClarity_r = \frac{1}{n}\sum_{i=1}^{n}(TC_{i+1} - TC_i)$$

where, TC_i represents the average task clarity of tasks deployed by a requester in the month i, n is the total number of months during which requester r deployed tasks.

Figure 7.6(f) presents our findings with respect to the overall change in task clarity corresponding to such requesters. The size of the points representing each requester depict the number of tasks deployed by that requester. We did not find a significant effect of the number of tasks deployed by requesters on the magnitude of change in task clarity.

Based on our findings, we understand that the overall task clarity in the marketplace fluctuates over time. We found a weak positive linear relationship between the number of tasks deployed by individual task requesters and the associated task clarity over time. However, we did not find evidence that the magnitude of change in task clarity is always positive in case of experienced requesters.

7.6.3 Top Requesters

We note that the top-3 task requesters accounted for around 67% of the tasks that were deployed between Oct'13 to Sep'14. The requesters were found to be *SpeechInk*–1,061 tasks, *AdTagger*–944 tasks, and *CastingWords*–824 tasks. The evolution of task clarity of the tasks corresponding to these requesters over time is presented in the Figure 7.7 below.

To understand the effect of the task requesters on the evolution of task clarity over time, we conducted a one-way between requesters ANOVA. We found a significant effect of task requesters on the evolution of task clarity across the three different requester conditions (SpeechInk, AdTagger, CastingWords) over the 12-month period; $F(2, 33) = 11.837$, $p < .001$. Post-hoc comparisons using the Tukey HSD test revealed that the evolution of task clarity corresponding to tasks from *SpeechInk* and *AdTagger* were significantly different in comparison to *CastingWords*.

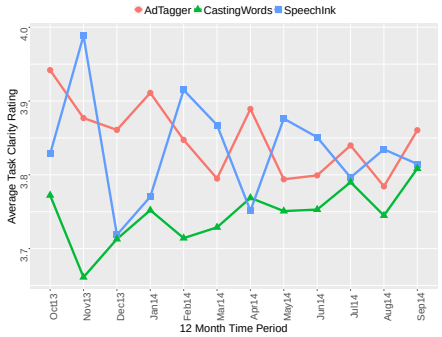


Figure 7.7: Top-3 task requesters w.r.t. the number of tasks deployed, and the evolution of their task clarity over time.

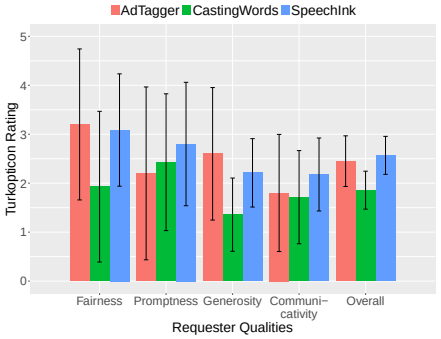


Figure 7.8: Average Turkopticon ratings of the top requesters from Oct'13–Sep'14.

We observe a gradual increase in the task clarity of *CastingWords* tasks over time in contrast to the other two top requesters. In the context of these requesters and the time period of Oct'13-Sep'14, we explored the Turkopticon ratings [105] corresponding to the requesters. Turkopticon collects ratings from workers on the following qualities : *fairness* of a requester in approving/rejecting work, *communicativity*– the responsiveness of a requester when contacted, *generosity*– quality of pay with respect to the amount of time required for task completion, *promptness* of the requester in approving work and paying the workers. Figure 7.8 presents a comparison of the Turkopticon ratings of the 3 requesters for each of the four qualities. We note that *SpeechInk* received consistently better ratings across all qualities within the given period. This coincides with the relatively higher task clarity of *SpeechInk* (*M=3.83, SD=0.47*) tasks when compared to *CastingWords* (*M=3.73, SD=0.48*) tasks over the 12 months (see Figure 7.7). A two-tailed T-test revealed a significant difference in the task clarity between *SpeechInk* and *CastingWords*; $t(1883)=18.43$, $p < .001$. We did not find ratings of tasks deployed by *AdTagger* on Turkopticon during the time period considered. However, we present a comparison based on the ratings received by *AdTagger* prior to Oct'13. Once again, in comparison to *CastingWords* we note that the higher overall quality ratings of *AdTagger* on Turkopticon coincide with the higher task clarity over the 12 months (*M=3.85, SD=0.48*); $t(1766)=25.23$, $p < .001$.

Through our findings it is clear that task clarity is not a global, but a local property of the AMT marketplace. It is influenced by the actors in

the marketplace (i.e., tasks, requesters and workers) and fluctuates with the changing market dynamics.

7.7 Conclusion

In this chapter we examined *task clarity*, an important, yet largely neglected aspect of crowdsourced microtasks. By surveying 100 workers, we found that workers confront unclear tasks on a regular basis. They deal with such tasks by either exerting extra effort to overcome the suboptimal clarity, or by executing them without a clear understanding. Poor task formulation thereby greatly hinders the progress of workers' in obtaining rewards, and in building up a good reputation.

To better understand how clarity is perceived by workers, we collected workers' assessments for 7.1K tasks sampled from a 5 years worth dataset of the AMT marketplace. With an extensive study we revealed that clarity is coherently perceived by workers, and that it varies by the task type. In addition, we found compelling evidence about the lack of direct correlation between clarity and complexity, showing the presence of a complex relationship that requires further investigation. We proposed a supervised machine learning model to predict task clarity and showed that clarity can be accurately predicted. We found that workers' perception of task clarity is most influenced by the number of keywords and title readability. Finally, through temporal analysis, we show that clarity is not a macro-property of the AMT ecosystem, but rather a local property influenced by tasks and requesters.

In conclusion, we demonstrated the importance of clarity as an explicit property of microwork crowdsourcing tasks, we proposed an automatic way to measure it, and we unveiled interesting relationships (or lack thereof) with syntactical and cognitive properties of tasks. Our findings enrich the current understanding of crowd work and bear important implications on structuring workflow. Predicting task clarity can assist workers in task selection and guide requesters in task design.

Part III

Task Assignment

This part addresses the problem of task assignment, i.e. associating tasks to crowds. Task assignment plays a centre role in knowledge creation systems, as it provides a mean to accelerate knowledge creation by optimizing the association of tasks to crowds, based on properties of both crowds and tasks. In this part, we formulate task assignment as a recommendation problem, i.e., we implement task assignment as recommending tasks to crowds. We will therefore use the term *task assignment* exchangeably with *task recommendation*, and use *users* and *items* in the context of recommendation to refer to *crowds* and (resources of) *tasks*, respectively. Building upon the previous chapters on crowd and task modeling, this part aims at showing how, by exploiting crowd and task properties, it is possible to design a new class of task assignment methods to improve the knowledge creation process.

In the following chapters, we will first demonstrate the benefit of incorporating crowd and task properties for task assignment. Then, we will introduce two novel methods that can exploit the structured nature of properties of crowds and tasks, which are often organized in a hierarchy (e.g. a taxonomy). Finally, we explore the state-of-the-art neural network technique to better learn the representations of crowds and tasks, thus to improve task assignment performance.

Chapter 8. By performing a study over 6 years of data from the StackOverflow platform, we first extend our previous study on crowd modeling to the intrinsic and extrinsic motivations, and their relationships across users and question topics. By defining metrics of expertise and (intrinsic and extrinsic) motivations, we show how they distribute and correlate across platform's users and topics. We then show how topic-specific combinations of motivations and expertise can help improve the accuracy of task assignment, thus accelerating the knowledge creation process.

Chapter 9. To fully exploit the properties of crowds and tasks for task recommendation, we investigate the relationship among the properties encoded in a hierarchical structure (e.g. a taxonomy). For this purpose, we study a specific class of recommendation methods, namely, feature-based recommendation. Feature-based methods mainly consider features (i.e. properties, in our context) that are organized in a flat structure where features are independent and in a same level. Through an empirical study, we show that *vertically*-affiliated features in a hierarchy can be used to describe the similarity between users or items. We thus propose a novel regularization method to model such similarity, namely, recursive regularization. Furthermore, we design a novel recommendation methods, namely ReMF, which integrates

recursive regularization into the widely used latent factor model to improve recommendation performance.

Chapter 10. We then extend our study to the relationships of *horizontally* organized features (i.e. siblings and cousins) in a hierarchy. We define metrics for two types of feature relationships that can be induced from the horizontal organization of features: complementary and alternative relationships. We show in real-world datasets that feature relationships in horizontal dimension can help explain and further model user-item interactions. To fully exploit feature hierarchies, we propose a unified recommendation framework, i.e. HieVH, to seamlessly fuse both vertical and horizontal dimensions for recommendation. Extensive validation on real-world datasets shows the superiority of HieVH against the state of the art.

Chapter 11. Finally we look into neural network based methods, which have shown to be highly effective in learning representations for a variety of objects, such as words in natural language, visual objects in images, and recently, users and items in recommender systems. Given the specific requirement of recommendation, i.e., recommending a ranked list of items to users, we first adapt the general neural network based representation learning method to enable personalized ranking for recommendation. Following the previous chapters, we design a unified Bayesian framework, namely MRLR, to integrate personalized ranking with the structured properties of users and items for recommendation. By extensive validation on real-world datasets, we show the benefit of MRLR in providing more accurate recommendations as well as more interpretable recommendation results.

To summarize, this part first demonstrates the efficacy of the properties of crowds and tasks for task assignment. By formulating task assignment as a recommendation problem, we further push forward the field of recommendation by contributing novel recommendation methods that can fully exploit the structure of properties of crowds and tasks.

Chapter 8

Harnessing Engagement for Knowledge Creation Acceleration in Community Q&A Systems

This chapter investigates the effect of incorporating properties of crowds and tasks for task assignment. Specifically, we focus on community question answering (CQA) systems, and study the effect of user engagement properties, including intrinsic and extrinsic motivations, and expertise, on question recommendation. We perform a cross-topic analysis, to show the positive effect of user properties and question topics on improving question recommendation performance.

This chapter is published as "Harnessing Engagement for Knowledge Creation Acceleration in Community Q&A Systems" [230], by J. Yang, A. Bozzon, and G.-J. Houben in Proceedings of the 23rd International Conference on User Modeling, Adaptation, and Personalization, pages 315-327. Springer, 2015.

8.1 Introduction

In community question-answering (CQA) systems users (askers) create questions, counting on topically-defined communities to provide an answer to their needs. Community members can browse existing questions, and decide whether or not to contribute to ongoing discussions. Such decisions are influenced by a multitude of factors, including time constraints, quality and difficulty of the question, and the knowledge of the answerer. Previous work [10, 229, 204] shows how engagement elements such as gamifications mechanisms, and expertise valorization can provide users with the right incentive for participation and collaborative knowledge creation.

While successful, such factors cannot prevent CQAs from facing several sustainability challenges. The volume of submitted questions overgrows the amount of new users willing, and capable, of answering them; a large portion of questions do not receive good (up-voted) answers, and even well-posed questions might wait for a long time before receiving a good answer [176, 228]. Recent studies have proposed to solve these problems with acceleration mechanisms such as: automatic detection of poorly formulated questions, question editing suggestion [176, 228], or question routing [84, 170, 40, 235].

To maximize the effectiveness of such mechanisms, a better comprehension of the mechanisms of knowledge creation in CQAs is needed. Recent research [85, 229] shows that engagement and topical expertise are complementary user properties. In this chapter we advocate for a more in-depth understanding of the interplay that exists between them, and we aim at demonstrating how they can be used to accelerate knowledge creation in CQA systems.

Our working hypothesis is that the process of knowledge creation is topically dependent, and that is driven by a mix of *intrinsic* motivations, *extrinsic* motivations, and *topical expertise* of CQAs users. We suggests that different topic-specific knowledge needs demand for different types of contributor: intuitively, to generate the best answer, some questions may require active answerers engaged in discussion; others may only need one expert user to directly provide the right answer. To test our hypothesis we focus on StackOverflow, a question-answering system specialized in programming-related issues. The chapter provides the following original contributions:

1. A study, focusing on the relation that exist between intrinsic motivations (e.g. interest), extrinsic motivations (e.g. reward), and expertise in topically-centred communities;

2. An off-line question routing experiment, aimed at verifying the impact of (intrinsic and extrinsic) motivations and expertise in user modelling for question recommendation.

Our work provides novel insights on the mechanisms that regulates knowledge creation in CQA systems. Although the study and the experiment focus on StackOverflow data, we believe that our results are of general interest. The study shows the relevant impact that different topics exercise on (intrinsic and extrinsic) motivation and expertise: the results can be used to devise novel engagement and retention mechanisms, aimed at accelerating knowledge creation by maximising the effectiveness of contributors. The experiment presented in the chapter provides empirical evidences of how existing CQAs can profit from the adoption of question routing mechanisms that include topical interest, motivations, and expertise as user modelling properties.

The remainder of the chapter is organized as follows: Section 8.2 briefly introduces engagement dimension in CQAs. Section 8.3 analyses (intrinsic and extrinsic) motivations and expertise in StackOverflow, while Section 8.4 shows how they can improve question routing performance. Section 8.5 describes related work, before Section 8.6 presents our conclusions.

8.2 Engagement Dimensions In CQA Systems

User engagement is defined as "the emotional, cognitive and behavioural connection that exists, at any point in time and possibly over time, between a user and a resource" [13]. Among the attributes that characterize engagement (e.g. aesthetics, endurability, novelty, reputation), *user context* embeds a combination of user- and context-dependent factors that profoundly influence and affect the relation between CQAs and their users. In this work we focus on two factors, namely: the *motivation*s driving users' activities; and users' *expertise*, as assessed by their peers, in a given topic of interest.

Motivation, is a precondition for action. To foster user engagement, system designers must understand the reasons why users take a particular action. The Self-Determination Theory [58] differentiates between *intrinsic* and *extrinsic* motivation.

Intrinsic motivations lead individuals to perform an activity because of their personal *interest* in it; or because its execution gives some form of *satisfaction*. Users of CQAs are often intrinsically motivated [161]; they decide to

interact with systems and their communities: *a*) To look for existing solutions to their issues; this involves browsing the CQA content, in search for the right formulation of knowledge need and, the answer(s) to it. *b*) To post a new question to the community, when no existing solution can be found. Or, *c*) to get satisfaction from the sense of efficacy perceived when, convinced to possess the skills and competence required to contribute to an ongoing discussion, they provide a new answer, or they comment/vote existing questions and answers.

When *extrinsically motivated*, individuals perform an activity for an outcome different from the activity itself, e.g. to obtain *external rewards*. A typical example of an engagement mechanism that exploits extrinsic motivation is *gamification* [10]. CQA systems often adopt two forms of external rewards: 1) a public *reputation score*, calculated by summing the number of votes obtained by all the posted questions and answers; and 2) a set of *badges*, assigned after achieving pre-defined goals (e.g. complete at least one review task, achieve a score of 100 or more for an answer).

Expertise. An expert can be defined as someone who is recognized to be skilful and/or knowledgable in some specific field [66], according to the judgment of the public, or of peers. In CQAs, social judgement is critical for expert identification. A question is usually answered by a set of users, whose answers are voted up or down by other members of the platform, thus reflecting the a user's capability of applying knowledge to solve problems. Hence, voting from other users can be seen as an unbiased, cyber simulation of social judgement for the answerers' expertise level [229]. Expertise can be seen as an example of intrinsic motivation related to competence. However, we stress the fundamental difference that exists between one's perception of competence (which is self-established, and often biased), and social judgement: by being externally attributed, the latter might not set off the same type of intrinsic triggers. Simply put: being perceived as an expert does not necessarily imply behaving like one. Next section elaborates on this behavioural difference, and provides quantitative support to our classification choice.

8.3 Analysing Extrinsic Motivations, Intrinsic Motivations, and Expertise in StackOverflow

The first part of our work studies how intrinsic motivations, extrinsic motivations, and expertise manifest themselves in topically-centred CQAs communities. We analyse StackOverflow, a popular CQA system launched in

2008 with the goal of becoming a very broad knowledge base for software developers. StackOverflow now features more than 2.7M users, 6.5M active questions, 11.5M answers, 26.1M comments, and 35.2K tags used by users to briefly characterize the subjects of the submitted questions. StackOverflow periodically releases a public version of the platform database, which can be accessed at https://archive.org/details/stackexchange. Our study is based on data created up until January 2014.

To investigate topical diversity, we categorize tags into 14 topics, shown in Table 8.1. Topics are identified by analysing the tag co-occurrence graph, using the approach described in [25].

8.3.1 Topical Influence on Extrinsic and Intrinsic Motivated Actions

Table 8.1 reports topical knowledge demand statistics. For each topic, we include: the number of submitted questions $\#Q$, as a measure of knowledge demand popularity; the number of answers $\#A$ and the number of comments C, as a measure of community participation. Comments or answers to self-created questions are not considered as extra contributors to the topic. Results highlight great topical diversity for both popularity and participation. It also emerges a topic-dependent distribution of answers and comments, which underlines differences in the type of activities performed by contributors.

The difference is more evident when observing the right-hand side of Table 8.1, which analyses communities' composition. $\#AU$ and $\#CU$ respectively indicate the number answerers and the number of commenters; $\#AU \cap \#CU$ shows the percentage of contributors who are both commenters and answerers; $\#CU - \#AU$ reports the percentage of contributors that are only commenters, and $\#AU - \#AU$ the percentage of contributors that are only answerers.

The distribution of contributors across topics greatly varies. We observe a general trend towards communities where the number of users acting exclusively as answerers is higher; together with an absolute higher number of answers, these figures suggest a preference for rewarded actions. In the `LAMP` topic the trend is more evident. `iOS` and `Adobe` are exceptions, as the percentage of users that exclusively comment is higher, and the absolute numbers of comments and answers is comparable. We observe no trend related to topics's popularity or participation. For instance, `Web`, `Java`, and `Databases` have a very similar number of commenters and answerers, which are mostly overlap-

	Knowledge Demand			Contributor Composition				
Topic	**Tags**	**#Q**	**#A**	**#CU**	**#AU**	**%(AU∩CU)**	**%(CU-AU)**	**%(AU-CU)**
.Net	c#, asp.net, .net, vb.net, wcf	571K	1222K	102K	119K	73.84%	6.40%	19.76%
Web	javascript, jquery, html, css	569K	1181K	146K	149K	85.43%	6.40%	8.17%
Java	android, java, eclipse	566K	1097K	136K	136K	85.84%	7.32%	6.84%
LAMP	php, mysql, arrays, apache	432K	927K	39K	128K	21.10%	6.98%	71.92%
C/C++	c, c++, windows, qt	269K	679K	78K	80K	87.45%	10.19%	12.36%
iOS	iphone, ios, objective-c	262K	441K	59K	57K	79.24%	11.98%	8.78%
Databases	sql, sql-server, database	177K	406K	74K	73K	79.37%	10.68%	9.95%
Python	python, django, list	186K	390K	55K	61K	67.59%	12.00%	20.41%
Ruby	ruby, ruby-on-rails	129K	226K	32K	39K	59.57%	12.89%	27.54%
String	regex, string, perl	99K	264K	47K	57K	57.26%	13.93%	28.81%
OOP	oop, image, performance, delphi	88K	212K	52K	61K	59.51%	14.20%	26.29%
MVC	asp.net-mvc, mvc	50K	98K	23K	29K	54.18%	15.06%	30.76%
Adobe	flex, flash, actionscript	39K	65K	18K	17K	73.98%	14.34%	11.68%
SCM	git, svn	34K	74K	21K	25K	44.31%	21.81%	33.88%

Table 8.1: Topical categorization of tags, with basic knowledge demand and contributors composition statistics.

ping. Other topics like `.Net`, `Python`, and `Ruby` show a slight predominance of answerers of comments, which is reflected in the uneven composition of contributors.

8.3.2 Measures of Motivations and Expertise in StackOverflow

Given its multi-faceted nature, user engagement has been measured in different ways: from subjective (e.g. user questionnaires) to objective (e.g. subjective perception of time) metrics, each measure is characterized by its own cost of acquisition, generalization capabilities, and bias.

In this work we consider several objective measures. As common in related literature, we define such measures over the set of StackOverflow users' activities available in the public dataset. Namely: posting new questions, answers, or comments; and voting existing ques

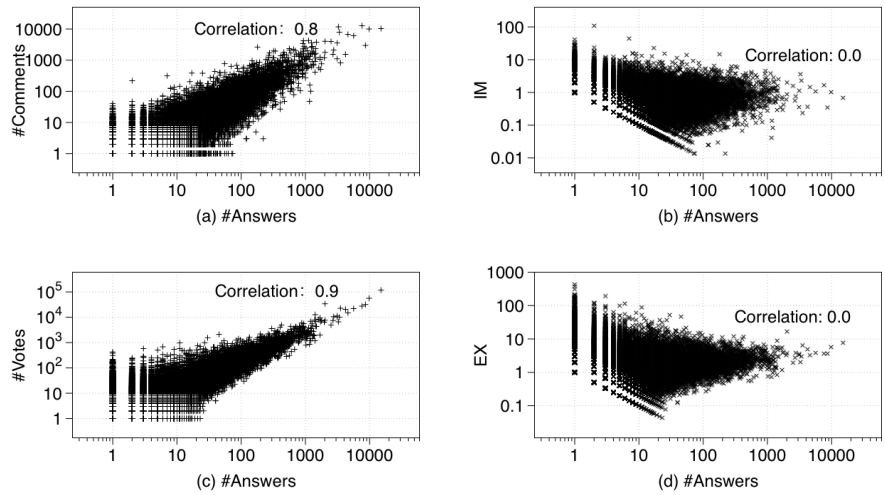


Figure 8.1: Distribution of number of comments and votes in the `.Net` without – (a) and (c) – and with – (b) and (d) – activeness correction.

Intrinsic motivations metric. The StackOverflow dataset does not provide page-access (view) data about individual users, thus making the task of measuring intrinsic motivations more challenging. To account for the missing data, we focus on *comments*, i.e. the only type of activity not rewarded by the scoring mechanism of StackOverflow. By being unrewarded, we assume commenting actions to be performed only for personal interest in a question,

in its topic, or in the community. Figure 8.1 (a) plots the distribution of comments and answers for each user participating in the `.Net` topic. As typical in StackOverflow [229], user activeness brings a strong bias (0.9 correlation, $p < .01$), as most active users are also more likely to engage in discussions, or provide minor help and criticisms. To compensate for the activeness bias, we use as intrinsic motivation measure $IM_u = \frac{\#C_u}{\#A_u}$, defined as the ratio between the number of comments and the number of answers provided by a user for a given set of topics. Intuitively, IN_u provides a measure of intrinsic motivation by quantifying the self-driven likelihood of a user to contribute to an ongoing discussion. Figure 8.1 (b) plots the distribution of IN_u, showing its independence from user activeness (0.0 correlation, $p < .01$).

Extrinsic motivations. External rewards such as reputation score and badges are strictly correlated with user activeness [229] which, in turns, is linearly correlated with the number of provided answers. We therefore use as a measure of extrinsic motivation $EM_u = \#A_u$, i.e. the number of answers provided by a user for questions about a a given set of topics, in a given time frame.

Expertise. In StackOverflow, social judgment in expressed in terms of votes assigned to questions or answers provided by users. The number of votes received by other users can be used as a measure of expertise. As for intrinsic motivations, most active users are also more likely to receive more votes for their contributions, as can be seen from Figure 8.1 (c) ($p < 0.01$). To normalize for user activeness, we use as expertise metric $EX_u = \frac{\#V_u}{\#A_u}$, i.e. the average number of votes received for each answer. Figure 8.1 (d) plots the distribution of EX_u ($p < 0.01$) for each user in the dataset.

8.3.3 Topical Relation Of Extrinsic Motivation, Extrinsic Motivation, and Expertise

Table 8.2 reports, for each topic and engagement metric defined in Section 8.3.2, the mean value (μ), standard deviation (σ) and skewness (γ) of their distributions.

The distributions of users' expertise (EX_u), intrinsic (IM_u) and extrinsic (EM_u) motivations is topically diverse. With metrics of motivation, a general trend can be observed: the averaged EM_u value for very popular topics (e.g., `.Net`) is higher than less popular ones (e.g., `SCM`), while the averaged value of IM_u is lower, meaning that users of these topics are, on average, active in providing answers to gain reputation while less self-interested participating to unrewarded activities. All metrics features very skewed distributions,

Topic	Basic Statistics ($\mu \pm \sigma, \gamma$)						Pearson Correlation		
	EX_u		IM_u		EM_u		EX_u-IM_u	EX_u-EM_u	IM_u-EM_u
.Net	1.65±4.73,	27.25	0.36±0.99,	18.57	10.23±84.17,	86.58	.03($p < .01$)	.02($p < .01$)	.05($p < .01$)
Web	2.06±9.55,	47.37	0.37±9.02,	7.01	7.90±52.35,	40.82	.02($p < .01$)	.00($p = .06$)	.08($p < .01$)
Java	2.30±9.69,	39.92	0.37±1.08,	21.85	8.10±64.61,	64.95	.01($p < .01$)	.00($p = .24$)	.01($p < .01$)
LAMP	1.59±6.09,	53.08	0.41±1.00,	10.42	7.25±44.05,	41.56	.02($p < .01$)	.01($p < .01$)	.08($p < .01$)
C/C++	2.01±6.76,	40.35	0.47±1.17,	9.30	8.41±58.29,	28.47	.03($p < .01$)	.02($p < .01$)	.09($p < .01$)
iOS	2.44±7.81,	18.98	0.38±1.08,	12.65	7.70±39.38,	19.11	.01($p < .01$)	.00($p = .35$)	.05($p < .01$)
Databases	1.69±7.11,	52.15	0.43±0.99,	6.57	5.53±40.94,	41.82	.01($p < .01$)	.01($p = .02$)	.05($p < .01$)
Python	2.43±6.68,	21.68	0.45±1.02,	6.52	6.44±75.77,	48.15	.03($p < .01$)	.02($p < .01$)	.06($p < .01$)
Ruby	2.68±8.44,	24.67	0.37±0.95,	8.27	5.88±27.23,	21.73	.02($p < .01$)	.01($p = .19$)	.06($p < .01$)
String	2.32±10.69,	51.44	0.57±1.21,	6.52	4.65±25.14,	31.61	.01($p < .01$)	.01($p < .01$)	.06($p < .01$)
OOP	2.18±7.54,	30.51	0.54±1.23,	8.18	3.47±16.43,	50.01	.04($p < .01$)	.02($p < .01$)	.07($p < .01$)
MVC	2.08±6.13,	22.34	0.40±0.95,	5.60	3.79±34.27,	132.39	.02($p < .01$)	.01($p = .19$)	.02($p < .01$)
Adobe	1.28±6.88,	77.62	0.24±0.71,	6.41	3.68±19.45,	36.33	.02($p = .03$)	.00($p = .97$)	.07($p < .01$)
SCM	5.51±28.48,	22.99	0.41±0.98,	6.01	2.99±23.64,	87.91	.01($p = .06$)	.00($p = .60$)	.03($p < .01$)

Table 8.2: Distribution and correlation of IM_u, EM_u, and EX_u values across topics.

especially EM_u: this indicates a general trend towards the identification of a small group of users possessing high motivation and/or expertise.

To investigate the relation between engagement factors, for each topic we consider the list of contributing users; we calculate their topical IM_u, EM_u, and EX_u values, and evaluate the pairwise Pearson correlation. Results are reported at the right-hand side of Table 8.2. Correlation is generally very low, mostly at high level of significance ($p < .01$). Overall, this result validates our choice of measures: in the reference dataset, the three engagement factors are independently observable. $IM_u - EM_u$ correlation is more evident, although still very diverse across topics (e.g. 0.09 in `C/C++`, 0.01 in `Java`). Interestingly, the (low) correlation between extrinsic motivation and expertise is highly not significant in 5 topics. We interpret such lack of statistical support as the result of more homogeneous expertise distributions among very active community members. The phenomenon affects topics at varying levels of popularity and participation, so community size doesn't appear to be a relevant factor. Further investigations are left to future work.

8.4 Exploiting Extrinsic Motivations, Intrinsic Motivations, and Expertise for Question Routing Optimization

In this section we provide empirical evidence of how (intrinsic and extrinsic) motivations and expertise can be exploited to improve the knowledge creation process. We employ question routing (i.e. recommendation of questions to the most suitable answerers) as knowledge acceleration mechanism, and compare the performance of different routing model configurations.

8.4.1 Data Preprocessing and Analysis

We split the dataset into two partitions. We build user profiles by considering actions executed up to Dec 31, 2012; we refer to this data partitions as the `Training` partition. Routing performance on question-answering are assessed of the `Testing` partition, which includes 1 year worth of user actions (from Jan 19^{th} 2014). To avoid cold-start problems, we consider only users that performed at least one action in both partitions. As our assessment includes a comparison of answerers rankings, we include in our experiment only questions with at least two answerers. Table 8.3 reports the resulting dataset figures.

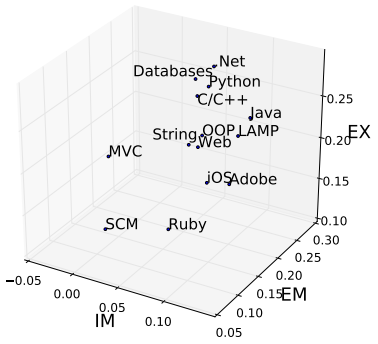


Figure 8.2: Pearson correlation of (intrinsic and extrinsic) motivations and expertise w.r.t. answer quality across topics.

	Train		Valid	
Topic	**#U**	**#Q**	**#U**	**#Q**
.Net	10,118	37,641	29,357	156,512
Web	13,877	51,267	34,034	180,230
Java	11,679	46,568	40,287	197,688
LAMP	11,305	35,079	35,070	149,487
C/C++	6,114	31,255	19,248	94,409
iOS	4,218	14,508	13,725	70,114
Databases	4,794	16,011	17,488	53,489
Python	4,988	18,380	15,227	55,546
Ruby	2,477	6,640	8,802	30,390
String	4,898	12,805	16,526	39,074
OOP	3,256	4,500	14,059	21,127
MVC	1,435	2,077	5,622	10,613
Adobe	182	267	1,649	4,703
SCM	814	1,546	3,871	7,275

Table 8.3: Users and questions distributions in the `Training`, `Validation`, `Testing` dataset partitions.

Our working hypothesis is that, by properly weighting different answerers according to their likelihood of being relevant to a given questions, the accuracy of question routing can be optimized. We test how the application of engagement factors in such weighting can lead to better question recommendation performance. To support our hypothesis, we first conduct the following experiment. For each question in the `Testing` set, we order answerers according to the number of votes they received from the community, and evaluate their intrinsic motivations, extrinsic motivations, and expertise measures over the `Training` set. We then calculate the Pearson rank correlations between the answering quality $AQ = \#votes$ and each of the three engagement factors (IM, EM, EX) : results are depicted in Figure 8.2. Each dot is a topic, and its coordinate indicate the respective correlations.

A higher correlation implies that the corresponding measure is more predictive for answering quality. The plot shows how, in general, intrinsic motivation is a poor predictor of answer quality, while EX and EM are more correlated, although often in a complementary fashion (e.g. `iOS`, `Adobe`). We observe great topical variety in the predictive power of the three engagement features. For instance, expertise in `Java` are more predictive than in `iOS`, while intrinsic motivation for these two topics are similar. Such a diversity calls for a routing model that weights the contributions of engagement properties differently across topics.

8.4.2 Routing Model

We propose a linear model, defined as follows:

$$S(u,q) = \alpha^t_{IM} IM_u + \alpha^t_{EM} EM_u + \alpha^t_{EX} EX_u,$$

For each question q of topic t in the `Testing` partition, $S(u,q)$ scores the answerer u's answer quality. $\alpha^t_{IM}, \alpha^t_{EM}, \alpha^t_{EX}$ respectively model the topic-specific needs for intrinsic motivated, extrinsic motivated, and expert answerers.

The optimal, topic-specific values for the $\alpha^t_{IM}, \alpha^t_{EM}, \alpha^t_{EX}$ parameters are calculated as follows. We identify a third dataset partition, called `Validation`, defined over the original, unfiltered dataset, and containing two years worth (from Jan 01, 2011 to Dec 31, 2013). Table 8.3 provides a basic description of the `Validation` partition, where the number of questions higher due to the lack of filtering conditions. We use the Linear Ordinal Regression SVM with L_2 regularization to learn the parameters, such that the profiled users are optimally ranked in the `Validation` partition. To learn more accurate parameters, we exclude the answer pairs in the training phase if the difference of #votes to the answers is less than 2. Such parameters are then used in the routing model to recommend questions to users in the `Testing` partition.

8.4.3 Experimental Setup

Evaluation Metrics The routing performance are assessed with three metrics, commonly used in the evaluation of recommender systems: NDCG (normalized discounted cumulative gain) [108], Kendall Tau, and Pearson rank correlation coefficients. The goal is to measure the quality/correlation of the recommended list of potential answerers by comparing it to the ground truth.

The evaluation of NDCG is performed against the #votes received by an answerer in a question. We use **NDCG@1** to assess the quality of the best recommended answerer, while **NDCG** assess the overall quality of the answerer set ranking. Due to the presence of negative voted answers, we exclude from the evaluation questions where the sum of DCGs is negative. Pearson correlation is calculated against #votes to answers, while Kendall Tau only measures similarity in the relative order of answers. Correlation is calculated only for questions where at least one answerer has a unique number of votes in the answer set.

Experimental Configuration. We compare the performance of 5 routing configurations. In the `Rdm` configurations, we randomly order the original answerers in the tested question. This configuration provides a performance baseline, as it measure a purely casual recommendation strategy. In the `Exp`, `Int`, and `Ext` configurations we respectively configure the routing system to return answerers according to their EX, IM and EM scores. These configurations simulate a recommendation strategy based on a single feature of engagement. Finally, `Cmb` applies the routing model described in Section 8.4.2, using the topic-specific learned parameters. As a remark, we exclude content-based model (e.g., bag-of-words of user answers) since our preliminary experiment show that it is less effective than configurations (e.g., `Ext`) that measure user answering activities.

8.4.4 Results

Table 8.4 summaries the results of our experiment. As expected, the topic of interest is an important performance diversification element for all the considered engagement factors and evaluation metrics. W.r.t. the numbers reported in Table 8.4, it is important to highlight how the range of values for NDCG metrics is necessarily narrower than for Pearson and Kendall Tau correlations. This is due to the definition of the metric which, by considering the number of votes received by an answerer, compress results in a more compact spectrum of values[2]. This is also demonstrated by the considerably high performance obtained by the `Rdm` configuration. Therefore, minor variations in NDCG values entails relevant differences in the quality of the returned answerers list.

As expected, among the configurations of `Int`, `Exp` and `Ext`, `Int` is the one providing worse results, whereas `Exp` configuration usually performs better than the others. On the other hand, we observe that `Cmb` configuration has in general performance better than or comparable with `Exp`. Small improvements, however, can provide tangible impacts. For instance, in topics such as `Web` and `iOS`, `Cmb` achieves better rankings – for 833 and 167 questions respectively. For some topics `Exp` could give even slightly better result than `Cmb` configuration, e.g., `Ruby, OOP`. Results suggests that the `Cmb` configuration could leverage different user engagement factors for question routing; however, is many topics, expertise is the most important factor for recommendation quality.

[2]**NDCG** = 1 entails a perfect recommendation.

As a final remark, we highlight how the routing performance of `Cmb`, `Exp` is generally higher than the ones reported in related literature [235]. This is despite the different targeted dataset, which is more extensive in our setting.

8.5 Related Work

This section positions this chapter in the context of previous work related to user engagement and knowledge creation acceleration in CQA systems.

Although both are factors of user engagement, user motivations and expertise in CQAs have been typically discussed in isolation. In a qualitative study based on interviews with CQA users, [161] finds that altruism, learning and competency are frequent motivations for participation. In addition, previous research shows gamification mechanisms can largely influence users' behaviours [25, 10]. Expertise, on the other hand, is mostly studied in the problem expertise identification. Related work typically adopts indicator-based methods such as Z_{score} [240], or graph-based methods such as the adapted PageRank method [137]. A recent study [229] shows how existing metrics of expertise can be heavily biased toward most active users. The normalization of user activeness in our EX_u metric is inspired by such consideration. Combining expertise and motivation, [166] explores their effect in the specific task of expert finding. W.r.t. literature our work further the understanding of user engagement factors in CQAs, providing new insights about the interplay of user expertise, intrinsic motivation, and extrinsic motivation. We contribute an original and extensive analysis that shows the independent manifestation of these three engagement factors across topical communities.

Knowledge creation acceleration is a topic recently emerged in research related to CQA systems. Typical methods include automatically detection of question quality [176], editing suggestions for poorly formulated questions [228], and active routing of questions to potentially relevant answerers [84, 170, 40, 235]. The latter is the most popular technique, and is the inspiration for our experiment. Previous work typically considers only users' topical activeness[84] as user modelling feature. These works were extended by considering the problem of routing question to a user community, for collaborative problem solving [170, 40]. More recently, [235] proposes a question routing user model that includes expertise, providing empirical evidences of its contribution to performance improvement. To the best of our knowledge, our work is the first considering a broader spectrum of engagement factors, and we extensively demonstrate their applicability.

	NDCG@1					NDCG					Pearson					Kendall				
Topic	Rdm	Exp	Int	Ext	Cmb	Rdm	Exp	Int	Ext	Cmb	Rdm	Exp	Int	Ext	Cmb	Rdm	Exp	Int	Ext	Cmb
.Net	.572	.687	.589	.676	**.693**	.834	.882	.842	.877	**.884**	.015	.279	.055	.244	**.290**	.014	.266	.054	.231	**.275**
Web	.578	.679	.624	.679	**.689**	.838	.879	.857	.878	**.883**	-.004	.234	.104	.225	**.255**	-.003	.226	.100	.217	**.245**
Java	.572	.665	.602	.647	**.666**	.835	**.873**	.847	.865	**.873**	.007	**.220**	.067	.169	.219	.005	**.210**	.064	.162	.209
LAMP	.579	.675	.602	.664	**.677**	.839	.877	.848	.873	**.878**	-.004	.219	.044	.193	**.228**	-.004	.212	.043	.187	**.220**
C/C++	.568	**.673**	.589	.644	.663	.834	**.878**	.843	.865	.874	.013	**.256**	.056	.181	.233	.015	**.244**	.052	.174	.222
iOS	.569	.644	.605	.650	**.658**	.835	.867	.850	.868	**.871**	-.002	.175	.080	.173	**.204**	.000	.171	.075	.169	**.197**
Databases	.593	.700	.614	.694	**.704**	.847	.889	.855	.886	**.890**	.001	.254	.037	.232	**.259**	.002	.248	.036	.228	**.254**
Python	.582	.682	.605	.684	**.695**	.842	.882	.851	.882	**.887**	.005	.244	.054	.236	**.265**	.005	.235	.052	.229	**.255**
Ruby	.607	**.656**	.628	.651	.651	.853	**.872**	.861	.870	.871	.016	**.141**	.073	.119	.130	.015	**.138**	.071	.119	.130
String	.572	.660	.601	.656	**.663**	.837	.874	.850	.872	**.875**	-.013	.200	.056	.171	**.206**	-.013	.192	.058	.165	**.196**
OOP	.578	**.682**	.614	.672	.680	.840	**.883**	.855	.879	.881	-.011	**.231**	.067	.185	.228	-.005	**.224**	.065	.185	.220
MVC	.623	.692	.613	.697	**.699**	.860	.888	.857	.890	**.890**	.034	.194	-.027	.193	**.214**	.034	.193	-.020	.199	**.208**
Adobe	.60	.663	.654	.649	**.674**	.853	.873	.872	.872	**.879**	-.013	.174	.96	.113	**.184**	-.009	.174	.97	.115	**.186**
SCM	.598	**.663**	.621	.650	.647	.853	**.875**	.861	.873	.871	-.038	**.101**	.010	.078	.083	-.044	**.100**	.008	.079	.081

Table 8.4: Experiment results of question routing with different configurations. Numbers in bold are the highest among all configurations.

8.6 Conclusion

The main mechanisms that drive knowledge creation process in CQAs are still to be fully uncovered. In this chapter we address the problem of characterising and measuring three engagement factors in The rationale behind our work is simple: to drive participation, thus improving the quality and speed of knowledge creations, we need to better understand the driving forces behind user engagement. Inspired by engagement theory from literature, we focus on intrinsic motivations, extrinsic motivations, and expertise. We propose three metrics, defined over the set of actions available to users, and we show how *topic* plays a major role in influencing them. We investigate the relations that exist among these three engagement factors, and demonstrate their independent and decomposable nature. A question routing optimization experiment confirms the relevant role that engagement can play in knowledge creation acceleration.

Chapter 9

Learning Hierarchical Feature Influence for Recommendation by Recursive Regularization

This chapter investigates structured properties of crowds and tasks for recommendation. We address the problem in the context of feature-based recommendation, and specifically focus on features organized in a hierarchy. We develop a novel regularization method, namely recursive regularization, to model the similarity of users or items induced from their vertically-affiliated features in a hierarchy. Moreover, we design a novel recommendation method, i.e. ReMF, that integrates recursive regularization to improve recommendation performance.

This chapter is published as "Learning Hierarchical Feature Influence for Recommendation by Recursive Regularization" [233], by J. Yang, Z. Sun, and J. Zhang in Proceedings of the10th ACM Conference on Recommender Systems, pages 51-58. ACM, 2016.

9.1 Introduction

Recommender systems aim to model user preferences towards items, and actively recommend relevant items to users. To address the *data sparsity* and *cold start* problems [183], feature-based recommendation methods, such as collective matrix factorization (CMF) [202], SVDFeature [42], and factorization machine (FM) [179], quickly gain prominence in recommender systems. They enable the integration of auxiliary features about users (e.g. gender, age) and items (e.g. category, content) with historical user-item interactions (e.g. ratings) to generate more accurate recommendations [199].

While commonly arranged in a flat structure (e.g. user gender, age), auxiliary features can be organized in a "feature scheme", i.e. a set of features that includes relationships between those features. Hierarchies are a natural yet powerful structure to human knowledge, and they provide a machine- and human- readable description of a set of features and their relationships. Typical examples of feature hierarchies include category hierarchy of on-line products (e.g. Amazon web store [151]), topic hierarchy of articles (e.g. Wikipedia [102]), genre hierarchy of music (e.g. Yahoo! Music), etc. The benefits brought by the explicit modeling of feature relationships through hierarchies have been investigated in a broad spectrum of disciplines, from machine learning [119, 109] to natural language processing [102]. How to effectively exploit feature hierarchies in recommender systems is still an open research question. We here provide a running example to illustrate how hierarchical feature structure can provide better recommendations.

Running Example. Consider a point of interest (POI) recommender system [236], where the goal is to recommend a real-world POI (e.g. a restaurant) to a user. User preferences can be influenced by geo-cultural factors: for instance, the country of residence might have an influence on user preferences for restaurants' cuisine (intuitively, an Italian and an American might have different culinary tastes). Likewise, fellow countrymen might show different preferences according to their city of residence (arguably, citizens from Boston and San Diego can show incredible different culinary preferences). This scenario is represented in Figure 9.1: users are described by auxiliary features that characterize their countries and cities of residence. These features are organized in a hierarchy, where cities are related to countries by a *locatedIn* relationship.

We consider the historical POI interactions in a given city of four users in Figure 9.1: Alice and Bob, from Rome and Florence (Italy); Charlie and Dave from Boston and San Diego (US). Italian users (Alice and Bob) both

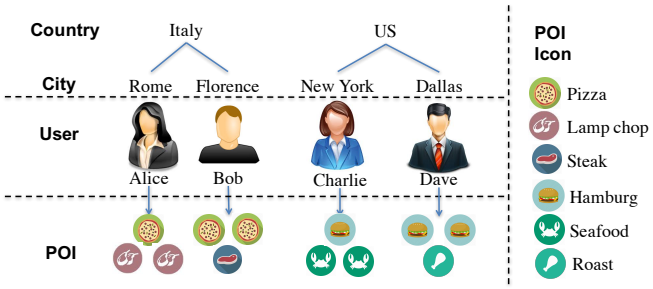


Figure 9.1: POI recommendation with auxiliary features hierarchy.

show preferences towards Pizza, thus suggesting a "national imprint" on their preferences. However, the two users are differently influenced by their country of residence, as Alice's preference for pizza is weaker than that of Bob: Alice checks in more at Lamp chop restaurants, while Bob checks in more at Pizza restaurants. A similar observation holds also for Charlie and Dave: the influence of US is weaker than that of Boston on Charlie, while stronger than that of San Diego on Dave.

The example highlights how related features (a country and its cities, linked by the *locatedIn* relationship) can co-influence user preferences, although the strength of the co-influence varies across relationship instances (e.g. Italy-Rome, Italy-Florence). This observation suggests the need for feature relationships (e.g. the *locatedIn* relationship) to be properly considered in recommendation methods. This co-influence could be known as a priori, but it is often best learnt from historical user-item interaction data.

Existing feature-based methods, e.g. SVDFeature [46], CMF [202] and FM [179], ignore the useful information provided by feature relationships, imposing a conversion step that transforms a hierarchical structure into a flat one. To fully exploit feature hierarchies, the main challenge is to model the co-influence of features on user-item interactions, determined by both the feature relationships in the hierarchical structure and the historical user-item interaction data.

Original Contribution. We propose a novel approach that *models* the co-influence of hierarchically-organized features on user-item interactions, and *learns* the strength of such co-influence from historical user-item interaction data, to improve recommendation performance. We first define the influence of an individual feature as regularization on latent factors, then combine the regularization of individual features by weighting them recursively over the hierarchy, from root to leaves, according to their organization. The regular-

ization of the feature hierarchy, named *recursive regularization*, is expressed as a regularization function parameterized by the weights associated to each feature. We then propose a novel recommendation framework `ReMF`, that integrates recursive regularization into the matrix factorization model to better learn latent factors. By learning the values of weights of each feature from the historical user-item interaction data, `ReMF` characterizes the influence of different features in a hierarchy on user-item interactions. We demonstrate the effectiveness of `ReMF` with an extensive validation performed on two recommendation scenarios, namely POI and product recommendation, and on multiple real-world data sets. Empirical results show that `ReMF` outperforms state-of-the-art approaches, scoring average improvements of 7.20% (MAE), 15.07% (RMSE) and 9.86% (AUC).

9.2 Related Work

Incorporating auxiliary features into recommendation, i.e. feature-based recommendation [183, 199], has become a popular and effective approach to address the *data sparsity* and *cold start* problems. A wide range of features has been explored, including user gender and age [4, 74], item category [123] and content [160, 76].

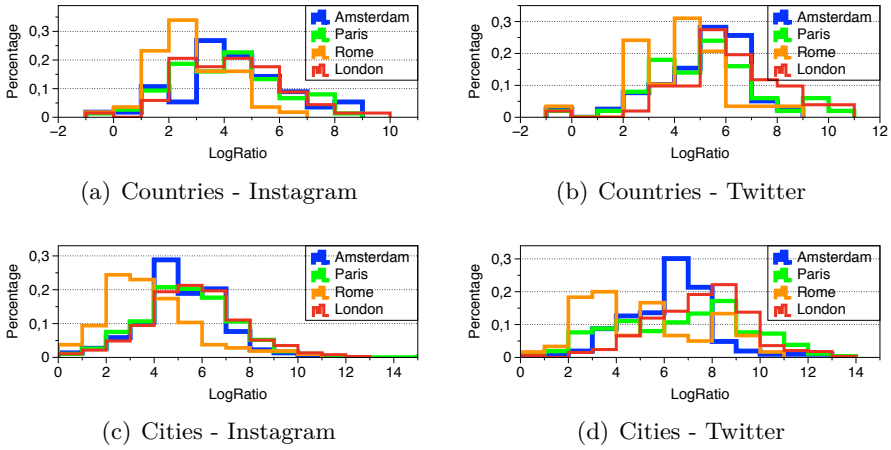
Many feature-based recommendation methods consider only features with a flat structure. For example, Singh et al. [202, 134] propose the collective matrix factorization (CMF) method, which factorizes the user-item rating and user-feature matrices simultaneously, to improve the recommendation performance. Chen et al. [42] devise a machine learning tookit, named SVDFeature. The basic idea is that a user's (an item's) latent factor is influenced by those of her (its) features. Rendle et al. [179, 181] design factorization machines (FM) that combines the advantages of Support Vector Machines with factorization model. However, all of these methods mentioned above cannot cope with hierarchical feature structure. Blending a feature hierarchy into these models requires converting the hierarchy into a flat structure, thus losing the information about feature relationships. To fully exploit a feature hierarchy, `ReMF` combines the distinct influence of different features on user-item interactions according to their structured relationships.

Some studies on *taxonomy*-aware recommendation incorporate hierarchy in recommendation. For example, Ziegler et al. [246] and Weng et al. [224] propose to model a user's taxonomy preferences as a flat vector, where each element corresponds to the user's preference over a taxonomy feature. The user's preference is modeled as the frequency the user rates items character-

ized by the feature. Albadvi et al. [6] propose a similar method, however it models each feature as a preference vector, where the elements are feature attributes (e.g. price, brand). All of these methods ignore feature relationships. Koenigstein et al. [123] design a new matrix factorization model for Yahoo! Music competition that incorporates the feature hierarchy of track album and artist. They predict user preferences by fusing item (e.g. track) latent factors with feature (e.g. album, artist) vectors. This idea is similar to SVDFeature [42]. Though feature relationships are considered, they cannot fully exploit a feature hierarchy as they simply add feature latent factors to item latent factors, without taking into account the dependent influence of hierarchically-organized features on user-item interactions.

Another related line of research focuses on integrating the structure *within* users/items in recommendation, e.g. social network [211, 143, 210], webpage network [132], tag network [243]. These methods usually regularize latent factors of users/items that are linked in the network, based on heuristic definitions of similarity between users/items. For instance, Ma et al. [143] propose SoReg that regularizes user latent factors based on cosine similarity of ratings between socially connected users. These methods consider the network *within* users/items, though can be applied in the case of feature hierarchy, e.g. by constructing implicit connections between users/items according to their feature relationships in the hierarchy. However, an essential difference between these methods and ours is that the influence of features on user-item interactions considered in these methods is usually hard-coded with manually defined similarity between users/items; on the contrary our proposed framework can automatically learn the co-influence of different features from the historical user-item interaction data. Recently Wang et al. [222] propose to model the implicit hierarchical structure *within* users and items based on user-item interactions. Our work differs from this one, in that we consider leveraging explicit auxiliary features to guide the learning of latent factors.

In summary, existing methods are incapable to model the co-influence of hierarchically-organized features on user-item interactions, thus restricting their applications in recommendation. In contrast, our framework can fully exploit an auxiliary feature hierarchy through the learning of hierarchical feature influence.



(a) Countries - Instagram (b) Countries - Twitter

(c) Cities - Instagram (d) Cities - Twitter

Figure 9.2: Distribution of the ratio between similarity (a, b) among countries and across countries, or (c, d) among cities and across cities, through the lens of (a, c) Instagram and (b, d) Twitter, including user visits to 4 European capital cities Amsterdam, Paris,Rome, and London (4 colors).

9.3 Data Analysis

This section demonstrates the need for recommendation methods able to account for the co-influence on user-item interactions of hierarchically-organized features.

Inspired by the running example, we show on 8 different data sets how: 1) users from the same country (named Country Visitors) are more similar in terms of POI preferences compared with users from different countries (named Foreign Visitors); and 2) users from the same city (named City Visitors) are more similar in terms of POI preferences compared to users from different cities *but the same country* (named Domestic Visitors).

Data Sets We collect data of Foursquare check-in's performed over 3 weeks in 4 European capital cities (Amsterdam, London, Paris, Rome) and published on 2 social media platforms (Twitter, Instagram). Table 9.1 shows the statistics about the 8 data sets. We consider users' residence *city*, *country* and *continent* as auxiliary information about users, as well as a root feature *residence location*. We use the method described in [23] to locate users' residence locations. For conciseness, we only analyze the co-influence of *country* and *city*. Overall we consider 121 countries and 2,873 cities.

Analysis Metrics. We denote all countries as $Con_1, \ldots, Con_s$; each country Con_s is the parent of all cities in it, i.e. $Con_s = parent(Cit_1, \ldots, Cit_t)$.

		Amsterdam	Rome	Paris	London
Inst.	#Users	4,318	4,081	11,345	12,719
	#POIs	5,768	7,878	14,849	12,892
	#Check-in's	28,142	26,714	80,553	66,092
	Sparsity	99.89%	99.92%	99.95%	99.96%
Twit.	#Users	1,599	1,369	6,521	9,305
	#POIs	3,816	4,876	16,046	14,117
	#Check-in's	8,670	8,727	43,541	48,852
	Sparsity	99.86%	99.87%	99.96%	99.96%

Table 9.1: Descriptive statistics of the data sets.

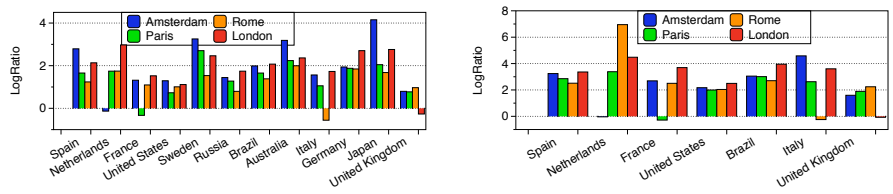
Each user u_i from a city Cit_t and a country Con_s ($Con_s = parent(Cit_t)$), has a set of visited POIs, i.e. $POI(u_i) = \{poi_{i1}, poi_{i2}, \ldots\}$. Then we measure the similarity between the users u_i and u_k using Jaccard similarity, i.e. $Jar(u_i, u_k) = |POI(u_i) \cap POI(u_k)|/|POI(u_i) \cup POI(u_k)|$.

We define u_i's similarity with the other Country Visitors (City Visitors), and with all Foreign Visitors (Domestic Visitors) as

$$Sim(F, u_i^w) = \frac{1}{|F| - 1} \sum_{u_k \in F, u_i \neq u_k} Jar(u_i, u_k),$$
$$Sim(F, u_i^a) = \frac{1}{|parent(F)| - |F|} \sum_{u_k \in parent(F), u_k \notin F} Jar(u_i, u_k),$$

respectively, where F is the country (or city) u_i resides in, and $|F|$ is the number of users characterized by the feature F. For instance, in the case of $F = Con_s$, the similarity between u_i and the other Country Visitors, denoted by $Sim(Con_s, u_i^w)$, is the averaged similarity between u_i and each of the other Country Visitors; the similarity between u_i and Foreign Visitors, denoted by $Sim(Con_s, u_i^a)$, is the averaged similarity between u_i and each of the Foreign Visitors. The similarity between u_i and the other City Visitors and Domestic Visitors can be similarly calculated. Now we define the overall similarity within a country (city) F, and across the country (city) and other countries (cities) as $Sim(F^w) = [Sim(F, u_1^w), Sim(F, u_2^w), \ldots]$ and $Sim(F^a) = [Sim(F, u_1^a), Sim(F, u_2^a), \ldots]$, respectively. Then we compare the overall similarity within a country (city), and that across the country (city) and other countries (cities) by:

$$LogRatio(F) = Log_2\Big(\frac{1}{|F|} \sum_{u_i \in F} \frac{Sim(F, u_i^w)}{Sim(F, u_i^a)}\Big),$$



(a) Countries - Instagram (b) Countries - Twitter

Figure 9.3: Ratio between similarity of users within a country and across the country and other countries, for countries with more than 100 cities observed through the lens of Instagram and Twitter.

where $LogRatio(F) > 0$ indicates that the elements in $Sim(F^w)$ is larger than that in $Sim(F^a)$ on average, and $LogRatio(F)) < 0$ otherwise. We test the significance of the difference between $Sim(F^w)$ and $Sim(F^a)$ with a Paired t-test.

Observation 1: *Country Visitors are more similar with each other in terms of POI preferences than with Foreign Visitors.*

The distribution of $LogRatio(Con)$ for all countries is shown in Figures 9.2(a-b) for Instagram and Twitter, respectively. More than 95% of the countries observed in both Instagram and Twitter have $LogRatio(Con) > 0$. Paired t-test shows that 95.88% countries in Instagram and 99.36% in Twitter have $Sim(Con^w)$ significantly larger than $Sim(Con^a)$ (p-value < 0.01). We thus conclude that Country Visitors are more similar with each other in terms of POI preferences than with Foreign Visitors.

Figures 9.3(a-b) show the $LogRatio(Con)$ for countries with more than 100 cities observed in the two platforms. We can see that users from different countries have different similarities when visiting the same city; and that the similarity of users from the same country varies across visited cities. These observations highlight the need for recommendation methods that can account for the variability caused by user residence country as well as visiting cities. Interestingly, all countries with $LogRatio(Con) < 0$ in both Figures 9.3(a,b) are the ones whose capital cities are visited, indicating that in visiting the capital city of their own countries, Country Visitors are less similar than Foreign Visitors. We find that this is due to that Country Visitors are mostly commuters in visiting their capital cities, i.e. they go to work places in the capital cities.

Observation 2: *City Visitors are more similar with each other in terms of POI preferences than with Domestic Visitors.*

The distribution of $LogRatio(Cit)$ for all cities in all countries is shown in Figures 9.2(c-d) for Instagram and Twitter. We can observe that all cities have $LogRatio(Cit)$ greater than 0; 88.44% of them in Instagram and 88.15% in Twitter have $Sim(Cit^w)$ significantly larger than $Sim(Cit^a)$ (p-value < 0.01). We therefore conclude that the similarity within City Visitors is higher than that with Domestic Visitors. Comparing the distribution of cities with that of countries, all cities have $LogRatio(Con) > 0$ while there are some countries with $LogRatio(Con) < 0$ (those whose capital cities are visited), indicating that users from the same city are more similar than users from the same country. Moreover, we find that generally cities have larger values of $LogRatio$ than countries. For instance the mean values of the distribution of Amsterdam in Figures 9.2(a,c) are 4.09 and 5.07, respectively. This observation hints that cities generally have larger influence than countries on their residents' preferences.

9.4 Recursive Regularization For Modeling Feature Co-Influence

We adopt the regularization technique to model the influence of auxiliary features. To do so, we have to consider feature relationships, and further allow for the learning of feature influence from historical user-item interaction data. For this we introduce a novel regularization method, named *recursive regularization*, that models the co-influence of features by recursively weighting each feature influence, traversing from root to leaves in the feature hierarchy.

9.4.1 Preliminaries

We first introduce the notations. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ be the set of m users, and $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ be the set of n items. Given a user-item interaction matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$, $\mathbf{R}_{ij}$ is a positive number denoting the rating given by u_i to v_j. $\mathbf{O} \in \mathbb{R}^{m \times n}$ denotes the indicator matrix, where $\mathbf{O}_{ij} = 1$ indicates that u_i rates v_j, and $\mathbf{O}_{ij} = 0$ otherwise. $\mathcal{F} = \{F_1, F_2, \ldots, F_t\}$ is the set of features, each of which describes at least one user in $\mathcal{U}$.

The features are organized hierarchically in a tree structure, where each node represents a feature in $\mathcal{F}$. The edge between a parent node $F_p \in \mathcal{F}$ and a child node $F_c \in \mathcal{F}$ represents a directed affiliation relationship, i.e. F_c

Notation	Explanation
$\mathcal{U}$,$\mathcal{V}$	user, item set
u_i/u_k, v_j	the ith/kth user in $\mathcal{U}$, and jth item in $\mathcal{V}$
$\mathbf{R}_{ij}$	rating given by user u_i to item v_j
$\hat{\mathbf{R}}_{ij}$	estimated rating for user u_i to item v_j
$\mathbf{O}$	indicator matrix indicating missing entries in $\mathbf{R}$
$\mathbf{U}_i$, $\mathbf{V}_j$	latent factors of user u_i and item v_j
$\mathcal{F}$	hierarchically-organized feature set
F	feature in the hierarchy
$Dis(F)$	regularization induced by *isolated* feature F
$Fu(F)$	feature unit with parent node F
$\mathbf{I}'(F)$	regularization by *isolated* feature unit $Fu(F)$
g, s	weighting parameters in propagating feature influence
$\mathbf{I}(F)$	regularization by feature unit $Fu(F)$ in hierarchy
$\mathbf{I}(\mathcal{F})$	regularization by feature hierarchy $\mathcal{F}$
$\mathbf{C}_{ik}$	regularization coefficient between $\mathbf{U}_i$ and $\mathbf{U}_k$
α	impact of recursive regularization
λ	regularizaton coefficient to avoid over-fitting
$\mathcal{J}$	objective function of ReMF framework

Table 9.2: Notations.

belongs to F_p. Figure 9.4(a) shows an example containing three leaf features F_1, F_2, F_3, i.e. features with no children. F_1, F_2 are children of the internal feature F_4. F_3 and F_4 are children of the root feature F_5. For simplicity, we assume that each user is explicitly associated with at most one leaf feature in $\mathcal{F}$. Table 9.2 summarizes all the notations throughout this chapter.

Our method is built on matrix factorization (MF) [125], which assumes the existence of latent structures in the user-item interaction matrix. By uncovering latent factors of users and items, it approximates the observed ratings and estimates the unobserved ratings. MF solves the following optimization problem:

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\sum_{i,j}\mathbf{O}_{ij}(\mathbf{R}_{ij}-\mathbf{U}_i\mathbf{V}_j^T)^2+\frac{\lambda}{2}(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2), \tag{9.1}$$

where $\mathbf{U}\in\mathbb{R}^{m\times d}$ and $\mathbf{V}\in\mathbb{R}^{n\times d}$ are the latent factors of users and items, respectively. d is the dimension of latent factors. λ is the regularization coefficient to avoid over-fitting. The unobserved rating for user u_i to item v_j can be estimated by the inner product of the corresponding user and item latent factors, i.e. $\hat{\mathbf{R}}_{ij}=\mathbf{U}_i^T\mathbf{V}_j$.

9.4.2 Modeling Influence of Feature Hierarchy on User-item Interactions

Step by step, we model the influence from a single feature to the combinations of features and finally the entire feature hierarchy.

Influence of an Isolated Feature. To start, we first define the regularization by an isolated feature F_p in the hierarchy as:

$$Dis(F_p) = \sum_{u_i, u_k \in F_p, i<k} \|\mathbf{U}_i - \mathbf{U}_k\|_F^2, \tag{9.2}$$

where $\|\mathbf{U}_i - \mathbf{U}_k\|_F^2$ is the squared Frobenius norm distance between the latent factors of u_i and u_k characterized by feature F_p: F_p poses regularization on the cumulation of the pairwise distance between users associated with it. Thus, $Dis(F_p)$ can be considered as the influence of the isolated feature F_p on user-item interactions by regularizing user latent factors. The definition here only considers the influence of an *isolated* feature, while the co-influence of the feature hierarchy contributed by the feature, i.e. influence of the feature *in the hierarchy*, is different from – but based on – the influence of the *isolated* feature, which will be illustrated later.

Our method models feature influence by regularizing user latent factors, and can be straightforward transferred to modeling the influence by regularizing item latent factors, or both of them.

Influence of an Isolated Feature Unit. Given the above definition, we now model the influence of an isolated combination of features, on learning user latent factors, by introducing the most important relationship among features in a hierarchy, i.e. *parent-child* relationship, based on which other relationships among features in the hierarchy such as *siblings*, *ancestors* can be derived. We first define the *feature unit*, i.e. $Fu(F_p)$, as the combination of a single parent node F_p and its children nodes, namely:

$$Fu(F_p) = \{F_p\} \cup \{F_c | \forall F_c \in children(F_p)\}.$$

Two examples of feature units $Fu(F_5)$ and $Fu(F_4)$ are shown in the red dash boxes in Figure 9.4(a).

Then we consider the influence of an isolated feature unit on learning user latent factors by regularization. For each isolated feature unit $Fu(F_p)$, we denote its influence as $\mathbf{I}'(F_p)$, and assign it two parameters g_p, s_p, with

the constraint $g_p + s_p = 1$. Parameters g_p and s_p are used to distribute the influence of the feature unit to two parts. One is given by the parent node, weighted by g_p, and the other is given by the children nodes, weighted by s_p. The influence of the isolated feature unit, i.e. $\mathbf{I}'(F_p)$, is then defined as:

$$\mathbf{I}'(F_p) = g_p Dis(F_p) + s_p(\sum_{\forall F_c \in children(F_p)} Dis(F_c)).$$

For example, the influence of the isolated feature unit $Fu(F_5)$ in Figure 9.4(a), i.e. $\mathbf{I}'(F_5)$, is determined by both the influence of the parent node F_5, i.e. $Dis(F_5)$, weighted by g_5, and the influence of its children nodes, i.e. $Dis(F_3)$ and $Dis(F_4)$, weighted by s_5. The overall influence of this isolated feature unit is: $\mathbf{I}'(F_5) = g_5 Dis(F_5) + s_5(Dis(F_3) + Dis(F_4))$. Compared with the influence of the isolated feature F_5, the influence of feature F_5 in $Fu(F_5)$ is different, in that $Dis(F_5)$ is weighted by g_5.

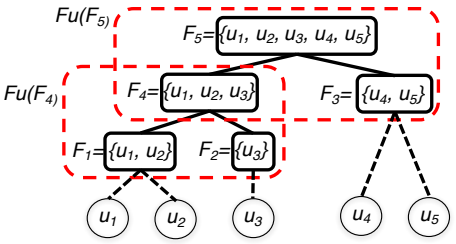
Influence of an Entire Feature Hierarchy. Based on the definition of the influence of an *isolated* feature unit, we now proceed to model the influence of feature unit *in the hierarchy*, thus to formally derive the overall influence of an entire feature hierarchy on user latent factors. Note that the influence of a feature unit *in the hierarchy* is different from – but based on – the influence of the *isolated* feature unit, and can be achieved by recursively defining the regularization of the feature unit *in the hierarchy*, given by:

Definition 1 (Recursive Regularization)

$$\mathbf{I}(F_p) = \begin{cases} g_p Dis(F_p) \quad +s_p(\sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c)), \\ \qquad\qquad \textit{if } F_p \textit{ is an internal feature}; \\ Dis(F_p), \qquad \textit{if } F_p \textit{ is a leaf feature and } |F_p| > 1; \\ 0, \qquad\qquad \textit{otherwise}, \end{cases}$$

where $|F_p|$ is the number of users characterized by feature F_p.

From the above definition, we can see the difference between the influence of a feature unit *in the hierarchy* $\mathbf{I}(F_p)$ and the influence of an *isolated* feature unit $\mathbf{I}'(F_p)$, that is, $\mathbf{I}(F_p)$ is recursively defined on $\mathbf{I}(F_c)$. Put another way, the influence of a child feature is included in the influence of its parent feature. Hence, the influence of an entire feature hierarchy, denoted by $\mathbf{I}(\mathcal{F})$, is equivalent to that of the root feature, as it recursively includes the influence of all features in the hierarchy. As an example, Equation 3 shows the influence of the feature hierarchy in Figure 9.4(a).



(a) Feature hierarchy

	u_1	u_2	u_3	u_4	u_5
u_1	-	1	$g_5+s_5g_4$	g_5	g_5
u_2	1	-	$g_5+s_5g_4$	g_5	g_5
u_3	$g_5+s_5g_4$	$g_5+s_5g_4$	-	g_5	g_5
u_4	g_5	g_5	g_5	-	1
u_5	g_5	g_5	g_5	1	-

(b) Regularization coefficients

Figure 9.4: (a) illustrates a feature hierarchy, where features with children (i.e. F_5, F_4) are called *internal features*. Particularly, F_5 is also named *root feature*, whereas features without children are called *leaf features*. Dash and solid lines respectively represent the user-feature (i.e. a user is characterized by a feature) and feature-feature (i.e. parent-child) relationships. Features in a red dash box comprises a feature unit. (b) shows the corresponding regularization coefficients of the corresponding example.

$$
\begin{aligned}
&\mathbf{I}(\mathcal{F}) = \mathbf{I}(F_5) \\
=&g_5 Dis(F_5) + s_5(\mathbf{I}(F_4) + \mathbf{I}(F_3)) \\
=&g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4(\mathbf{I}(F_1) + \mathbf{I}(F_2)) + Dis(F_3)) \\
=&g_5 Dis(F_5) + s_5(g_4 Dis(F_4) + s_4 Dis(F_1) + Dis(F_3)). \qquad (3)
\end{aligned}
$$

The deduction of recursive regularization of a feature hierarchy is shown in Algorithm 1, where the co-influence of features is modeled as a regularization function parameterized by the weights of each feature in the hierarchy. These weights characterize the influence of distinct features, and can be further learnt from historical user-item interaction data, as we introduce in the next section.

Remark. By recursively weighting and combining feature influence over a hierarchy from the root feature to the leaves, recursive regularization can model the influence of an arbitrarily deep feature hierarchy that can be either balanced or imbalanced.

Algorithm 1: Recursive Regularization Deduction

Input: feature hierarchy $\mathcal{F}$, $g_p, s_p \forall F_p \in \mathcal{F}$
foreach $F_p \in \mathcal{F}$ **do**
  $\mathbf{I}(F_p) \leftarrow 0$;
$layer \leftarrow$ #layers of $\mathcal{F}$;
for $l = 0; l \leq layer; l++$ **do**
  foreach *feature* F_p *at layer* l *of* $\mathcal{F}$ **do**
    if F_p *is a leaf feature* ($l = 0$) *and* $|F_p| > 1$ **then**
      $\mathbf{I}(F_p) \leftarrow Dis(F_p)$;
    else if F_p *is an internal feature* ($l \neq 0$) **then**
      $\mathbf{I}(F_p) \leftarrow g_p Dis(F_p) + s_p(\sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c))$;
$\mathbf{I}(\mathcal{F}) \leftarrow \mathbf{I}(F_{root})$;

9.5 ReMF: a Recommendation Framework Integrated with Recursive Regularization

We first introduce a novel recommendation framework `ReMF`, that integrates the recursive regularization into the MF model to exploit auxiliary feature hierarchy. Then an optimization method and the complexity analysis for `ReMF` are presented.

9.5.1 The ReMF Framework

By incorporating recursive regularization into the MF, the `ReMF` framework is defined by:

Definition 2 (The ReMF Framework)

$$\min_{\substack{\mathbf{U},\mathbf{V},\\ g_p,s_p \forall F_p \in \mathcal{F}}} \mathcal{J} = \frac{1}{2}\sum_{i,j} \mathbf{O}_{ij}(\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)^2 + \frac{\alpha}{2}\mathbf{I}(\mathcal{F}) + \frac{\lambda}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2)$$

where α is a regularization parameter that controls the impact of recursive regularization, i.e. $\mathbf{I}(\mathcal{F})$.

Thanks to recursive regularization, `ReMF` can model the co-influence of features in the hierarchy to learn user latent factors.

It also characterizes the distinct influence of each feature, thus helping with the interpretation of the effect of each feature in the hierarchy on recommendation, illustrated as follows.

Considering the example of Figure 9.4, based on Equations 9.2 and 3, the feature hierarchy influence $\mathbf{I}(\mathcal{F})$ can be rewritten as:

$$(g_5 + s_5 g_4 + s_5 s_4)\|\mathbf{U}_1 - \mathbf{U}_2\|_F^2 + (g_5 + s_5 g_4)\|\mathbf{U}_1 - \mathbf{U}_3\|_F^2 + \ldots,$$

where the strength of the regularization between u_1, u_2's latent factors is $(g_5 + s_5 g_4 + s_5 s_4)$, and that of u_1, u_3's latent factors is $(g_5 + s_5 g_4)$. In fact, the strength of regularization is the combination of influence of different features. For simplicity, we assume $g = s = 0.5$ for each internal feature. Therefore, the strength of regularization between u_1, u_2's latent factors is $(g_5 + s_5 g_4 + s_5 s_4) = 1$, from which we could see that the feature F_5 has an influence of $g_5 = 0.5$, while its children features F_4 and F_1 have influence of $s_5 g_4 = 0.25$ and $s_5 s4 = 0.25$, respectively. Then, for u_1, u_3, the strength of regularization between their latent factors is $(g_5 + s_5 g_4) = 0.75$, where the features F_5, F_4 have influence of $g_5 = 0.5, s_5 g_4 = 0.25$, respectively. The distinct influence of features on learning user latent factors can therefore be characterized by certain functions of the weights (g, s).

To formally derive feature influence on an arbitrary pair of users, we define the *regularization coefficient* $\mathbf{C}_{ik}$ to represent the strength of regularization between u_i and u_k, where a greater value of $\mathbf{C}_{ik}$ indicates a higher correlation between the two users. Hence, $\mathbf{I}(\mathcal{F})$ can be reformulated as:

$$\mathbf{I}(\mathcal{F}) = \sum_{u_i, u_k \in \mathcal{U}, i<k} \mathbf{C}_{ik}\|\mathbf{U}_i - \mathbf{U}_k\|_F^2,$$

We next introduce two theorems for deriving $\mathbf{C}_{ik}$, which is the combination of the influence by different features on u_i and u_k.

Theorem 1 *The regularization coefficient for any pair of users u_i, u_k (i.e. $\mathbf{C}_{ik}$) characterized by the same leaf feature is 1:*

$$g_{root} + s_{root}(g_{c_1} + s_{c_1}(g_{c_2} + s_{c_2}(...(g_{c_l} + s_{c_l})))) = 1,$$

where the list $\{F_{root}, F_{c_1}, F_{c_2}, \ldots, F_{c_l}\}$ is the set of the common features of u_i and u_k, ordered in a sequence from the root feature F_{root} to the leaf feature F_{c_l}.

Proof. This is straightforward to prove, due to the constraint $g + s = 1$. Considering the example $\{u_1, u_2\}$ in Figure 9.4, the sum of the relevant regu-

larization terms, i.e. $g_5 Dis(F_5), s_5 g_4 Dis(F_4)$ and $s_5 s_4 Dis(F_1)$, in Equation 3 is:

$$\begin{aligned}&(g_5 + s_5(g_4 + s_4))\|\mathbf{U}_1 - \mathbf{U}_2\|_F^2 \\ =&(g_5 + s_5)\|\mathbf{U}_1 - \mathbf{U}_2\|_F^2 = \|\mathbf{U}_1 - \mathbf{U}_2\|_F^2.\end{aligned}$$

Theorem 2 *For any pair of users u_i, u_k not characterized by a common leaf feature, the regularization coefficient (i.e. $\mathbf{C}_{ik}$) is:*

$$g_{root} + s_{root}(g_{c_1} + s_{c_1}(g_{c_2} + s_{c_2}(...(g_{c_l})))),$$

where the list $\{F_{root}, F_{c_1}, F_{c_2}, \ldots, F_{c_l}\}$ is the set of the common features of u_i and u_k, ordered from the root feature F_{root} to the deepest common feature F_{c_l}.

Proof. All possible features that can influence the regularization coefficient of u_i, u_k are their deepest common feature, and the parents and ancestors of the deepest common feature.

According to the above theorems, the value of regularization coefficient always falls into the range of $[0, 1]$, with 1 indicating the full regularization and 0 indicating no regularization. As an example, Figure 9.4(b) shows the regularization coefficients of the feature hierarchy in Figure 9.4(a).

These regularization coefficients naturally connect `ReMF` to network based recommendation methods, which also consider pairwise regularization on users. There are however two essential differences: 1) network-based regularization coefficients are usually hard-coded, while our regularization coefficients are modeled from the feature hierarchy structure, and expressed by the function of weights (g, s). And, 2) (g, s), which parametrizes the distinct feature influence while being automatically learnt from the historical user-item interaction data, as we will address in the next subsection.

9.5.2 The Optimization Method for ReMF

We adopt the stochastic gradient descent scheme [124, 125] to optimize our objective function.

Updating $\mathbf{U}, \mathbf{V}$. The gradients of $\mathbf{U}_i, \mathbf{V}_j$ are given by:

Algorithm 2: ReMF Model Learning

Input: rating matrix $\mathbf{R}$, feature hierarchy $\mathcal{F}$, $d, \gamma, \lambda, \alpha, iter$

Initialize $\mathbf{U}, \mathbf{V}, g_p, s_p$, and $\forall F_p \in \mathcal{F}$;
for $t = 1; t \leq iter; t++$ **do**
  foreach $\mathbf{U}_i \in \mathbf{U}$, $\mathbf{V}_j \in \mathbf{V}$ **do**
    $\mathbf{U}_i^{(t)} \leftarrow \mathbf{U}_i^{(t-1)} - \gamma \frac{\partial \mathcal{J}}{\partial \mathbf{U}_i}$;
    $\mathbf{V}_j^{(t)} \leftarrow \mathbf{V}_j^{(t-1)} - \gamma \frac{\partial \mathcal{J}}{\partial \mathbf{V}_j}$;
  foreach *Internal feature in the hierarchy* **do**
    $g_p^{(t)} \leftarrow g_p^{(t-1)} - \gamma \frac{\partial \mathcal{J}}{\partial g_p}$;
    $s_p^{(t)} \leftarrow s_p^{(t-1)} - \gamma \frac{\partial \mathcal{J}}{\partial s_p}$;
  Calculate $\mathcal{J}$ by Algorithm 1 and Definition 2;
  if $\mathcal{J}$ *has converged* **then**
    break;

$$\frac{\partial \mathcal{J}}{\partial \mathbf{U}_i} = -\sum_j \mathbf{O}_{ij}(\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)\mathbf{V}_j + \lambda \mathbf{U}_i + \alpha \sum_{u_i, u_k \in \mathcal{U}, i<k} \mathbf{C}_{ik}(\mathbf{U}_i - \mathbf{U}_k),$$
$$\frac{\partial \mathcal{J}}{\partial \mathbf{V}_j} = -\sum_i \mathbf{O}_{ij}(\mathbf{R}_{ij} - \mathbf{U}_i \mathbf{V}_j^T)\mathbf{U}_i + \lambda \mathbf{V}_j.$$

Updating (g, s)**.** (g, s) can be predefined heuristically, or handcrafted by domain experts who can fairly quantify the influence of different features. Instead, we provide an effective data-driven solution that automatically learns (g, s) based on the historical user-item interaction data.

We only need to estimate (g, s) for internal features in the hierarchy, since the leaf features do not have children. For an internal feature F_p, the gradients of g_p, s_p are equivalent to the multipliers of g_p, s_p in $\mathbf{I}(\mathcal{F})$. Thus, we have:

$$\frac{\partial \mathcal{J}}{\partial g_p} = \begin{cases} Dis(F_p), & \text{if } F_p \text{ is root,} \\ \prod_{\forall a: F_a \in ancestors(F_p)} s_a Dis(F_p), & \text{otherwise;} \end{cases}$$
$$\frac{\partial \mathcal{J}}{\partial s_p} = \begin{cases} \sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c), & \text{if } F_p \text{ is root,} \\ \prod_{\forall a: F_a \in ancestors(F_p)} s_a (\sum_{\forall F_c \in children(F_p)} \mathbf{I}(F_c)), & \text{otherwise.} \end{cases}$$

According to the constraint $g_p + s_p = 1$, we can update g_p (or s_p) using the gradient and the other by $s_p = 1 - g_p$ (or $g_p = 1 - s_p$). The detailed learning process is shown in Algorithm 2.

Complexity Analysis The computational time is mainly taken by evaluating the objective function $\mathcal{J}$ and updating the related variables. The time to compute the $\mathcal{J}$ is $\mathcal{O}(d|\mathbf{R}| + dm^2)$, where $|\mathbf{R}|$ is the number of non-zero observations in the rating matrix $\mathbf{R}$. For all gradients $\frac{\partial \mathcal{J}}{\partial \mathbf{U}_i}$, $\frac{\partial \mathcal{J}}{\partial \mathbf{V}_j}$, $\frac{\partial \mathcal{J}}{\partial g_p}$, $\frac{\partial \mathcal{J}}{\partial s_p}$, the computational time are $\mathcal{O}(d|\mathbf{R}| + dm^2)$, $\mathcal{O}(d|\mathbf{R}|)$, $\mathcal{O}\left(d\sum_{l=0}^{layer-1}\frac{\overline{m}_l(\overline{m}_l-1)n_l}{2}\right)$ and $\mathcal{O}(|s_p|)$, respectively. Wherein $\overline{m}_l$ denotes the average number of users in each node at layer l, n_l denotes the number of nodes at layer l, and $|s_p|$ ($\ll |\mathbf{R}|$) denotes the number of internal nodes. Particularly, we leverage $s_p = (1 - g_p)$ to update s_p. The overall computational complexity of Algorithm 2 is $(\#iteration * \mathcal{O}(d|\mathbf{R}| + dq))$, where $q = max(\sum_{l=0}^{layer-1}\frac{\overline{m}_l(\overline{m}_l-1)n_l}{2}, m^2)$. In real-world applications $\overline{m}_l$ is typically small (e.g. power-law distributed), thus making `ReMF` scalable to large data set.

9.6 Experiments and Results

We assess the performance of `ReMF` with a comparison with the state-of-the-art, feature-based, hierarchy-based recommendation methods. The comparison is performed over 1) the data sets introduced in Section 3, for POI recommendation with user feature hierarchy; and 2) a data set from the Amazon Web store [151], for product recommendation with item feature hierarchy.

9.6.1 Experimental Setup

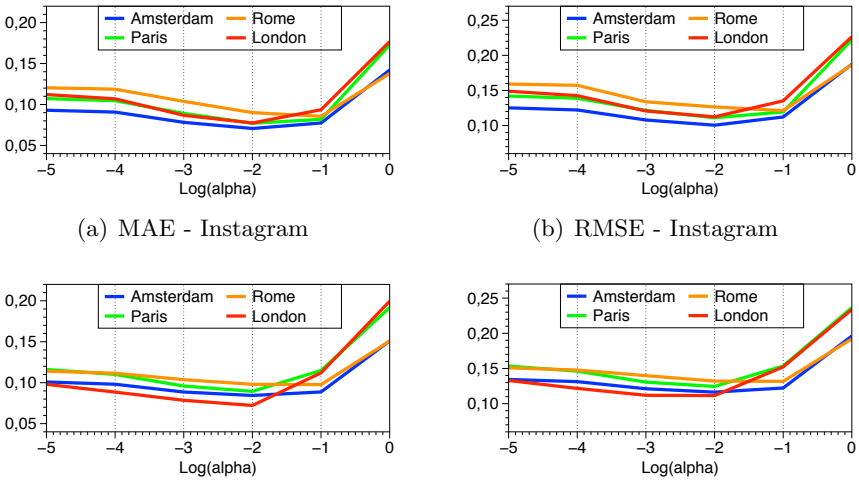
Evaluation. We adopt the standard 5-fold cross-validation, and the following 3 metrics for evaluation: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [123, 181] to measure the error of predicted ratings; and Area Under the ROC Curve (AUC) [97, 242] to measure the quality of predicted ranking of items (ranked according to the predicted ratings). The smaller MAE and RMSE, and the larger AUC, the better the recommendation performance.

Comparison Methods. The following methods are compared: (1) **MF** [125]: matrix factorization method; (2) **CMF** [202]: collective MF; (3) **TaxMF** [123]: taxonomy-based MF; (4) **SoReg** [143]: network-based recommendation method

incorporating social relations; (5) **FM** [181]: factorization machine; (6) **HieFM**: factorization machine with hierarchy information.

HieFM is a variation of FM that considers each features path in the hierarchy (from root to leaf nodes) as an additional feature in the design vectors of FM. Similar to FM, CMF and TaxMF can also incorporate path-based features. As FM outperforms CMF and TaxMF (see Section 9.6.3), we limit our comparison with previous methods exploiting path-based features to HieFM.

Parameter Settings. We empirically set optimal parameters for each method using a grid search in $\{0.0001, 0.001, 0.01, 0.05\}$ for both λ (including 1-way and 2-way regularization of FM) and the learning rate γ; $\alpha = 0.5$ for CMF; $\beta = 0.01$ for SoReg. For fair comparison, we set $d = 10$ (the dimension of latent factors) for all the methods, and adopt all features (i.e. continent, country, and city) as input in TaxMF, CMF, FM and HieFM. HieFM has path-based features as additional hierarchy information. In SoReg, we model the social relations among users by counting the number of common features, under the assumption that the commonality establishes implicit social relationships based on the geo-social correlation phenomenon [75]. Without loss of generality, we adopt $f(x) = 1/(1 + x^{-1})$ to map each #check-in $\mathbf{R}_{ij} \in \mathbf{R}$ in POI data sets into the interval $(0, 1)$ [76].



(a) MAE - Instagram (b) RMSE - Instagram

(c) MAE - Twitter (d) RMSE - Twitter

Figure 9.5: The effects of α on the performance of `ReMF` on Instagram and Twitter measured by MAE and RMSE.

9.6.2 Results of ReMF

We analyze the influence of recursive regularization on `ReMF` performance, and discuss how the weighting parameters g, s can help the interpretation of recommendation results.

The Impact of α. In `ReMF`, α controls the strength of recursive regularization of feature hierarchy. We apply a grid search in $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ to investigate the impact of α on recommendation performance. Results are shown in Figure 9.5. As α varies from small to large, the performance first increases then decreases, with the maximum reached at the range $[10^{-2}, 10^{-1}]$. The performance variations across data sets suggest the need for data set-specific settings; the similarity in performance variation across α values shows the robustness of `ReMF`.

Interpretation from (g, s). We examine (g, s) for the internal features, i.e. continents and countries, learnt from data. Table 9.3 shows the list of continents and countries ranked according to their g values. Recall that for a continent (country), $g > s$ means that the continent (country) has a stronger effect on user preferences than and its children features, i.e. countries (cities).

In general the continents have relatively smaller effects on user preferences (with g values all below 0.2), suggesting that continents have weaker effects than their countries. In addition, we observe a big variance in the g values of countries, indicating that different countries have different influence on user preferences. The high variance of countries' g values proves the necessity of parameterizing g, s in recommendation. We then compare the influence of countries and cities on their residents' preferences. As cities of a country and the country comprise a feature unit, the influence of a city can be measured by $s = 1 - g$, where g is the influence of the country. We can see from Table 9.3 that most countries have $g < 0.5$ (only 3 countries have $g > 0.5$), i.e. $s > 0.5$, indicating that the influence of cities in most countries have more influence on their residents' preferences than the countries themselves.

Continents		**Top countries**		**Bottom countries**	
Name	g	Name	g	Name	g
Europe	0.1837	Portugal	0.6915	Chile	0.0211
America	0.1656	Monaco	0.5813	Thailand	0.0175
Asia	0.1534	Serbia	0.5130	Spain	0.0100
Africa	0.0375	Poland	0.4453	Indonesia	0.0081
Oceania	0.0139	Hungary	0.4141	Belgium	0.0064

Table 9.3: Values of g for continents and top/bottom countries in the dataset.

View		Data Set	MAE							RMSE						
			MF	**CMF**	**TaxMF**	**SoReg**	**FM**	**HieFM**	**ReMF**	**MF**	**CMF**	**TaxMF**	**SoReg**	**FM**	**HieFM**	**ReMF**
All	Inst.	Amsterdam	0.1957	0.1564	0.1426	0.1038	0.0876	0.0822*	**0.0707**	0.3134	0.1940	0.1934	0.1455	0.1373	0.1352*	**0.1005**
		Paris	0.1539	0.1550	0.1416	0.1208	0.0790*	0.0830	**0.0772**	0.2675	0.1921	0.1849	0.1825	0.1293	0.1184*	**0.1111**
		Rome	0.2549	0.1584	0.1474	0.1355	0.0912	0.0885*	**0.0855**	0.3860	0.1967	0.1859	0.1912	0.1403	0.1389*	**0.1212**
		London	0.1799	0.1559	0.1369	0.1250	0.0834*	0.0851	**0.0774**	0.2964	0.1934	0.1762	0.1840	0.1347*	0.1396	**0.1124**
	Twit.	Amsterdam	0.2264	0.1606	0.1345	0.1229	0.0989	0.0942*	**0.0844**	0.3473	0.1996	0.1717	0.1669	0.1540	0.1454*	**0.1164**
		Paris	0.2014	0.1714	0.1552	0.1266	0.0956	0.0935*	**0.0894**	0.3207	0.2136	0.2038	0.1687	0.1408	0.1387*	**0.1245**
		Rome	0.2681	0.1713	0.1591	0.1345	0.1023	0.0996*	**0.0977**	0.3902	0.2132	0.2030	0.1831	0.1534	0.1469*	**0.1317**
		London	0.2176	0.1659	0.1545	0.1122	0.0931	0.0898*	**0.0772**	0.3075	0.2065	0.1959	0.1540	0.1407	0.1375*	**0.1115**
Cold start	Inst.	Amsterdam	0.2938	0.1552	0.1457	0.1051	0.0924	0.0885*	**0.0712**	0.3877	0.1926	0.1904	0.1479	0.1443	0.1391*	**0.1040**
		Paris	0.1939	0.1541	0.1476	0.1173	0.0849	0.0848*	**0.0799**	0.3110	0.1907	0.1896	0.1713	0.1374	0.1287*	**0.1183**
		Rome	0.3840	0.1614	0.1518	0.1356	0.0952	0.0938*	**0.0808**	0.4868	0.1990	0.1925	0.1845	0.1455*	0.1506	**0.1250**
		London	0.3032	0.1544	0.1415	0.1221	0.0893*	0.0901	**0.0791**	0.3978	0.1917	0.1819	0.1685	0.1426	0.1425*	**0.1161**
	Twit.	Amsterdam	0.3261	0.1604	0.1426	0.1189	0.1006	0.0966*	**0.0849**	0.4003	0.1982	0.1832	0.1609	0.1558	0.1514*	**0.1172**
		Paris	0.2439	0.1706	0.1640	0.1271	0.1012	0.0945*	**0.0873**	0.3764	0.2123	0.2120	0.1713	0.1485*	0.1502	**0.1226**
		Rome	0.3922	0.1718	0.1681	0.1343	0.1073	0.1070*	**0.0988**	0.4951	0.2133	0.2136	0.1833	0.1559	0.1517*	**0.1359**
		London	0.3301	0.1642	0.1587	0.1128	0.0967	0.0924*	**0.0756**	0.3976	0.2043	0.2013	0.1563	0.1475	0.1436*	**0.1093**

Table 9.4: Performance of the considered recommendation methods on the testing views "All" and "Cold start" of POI data sets. The best performance for each city is boldfaced; the runner up is labelled with "*". The improvements by the best method on all data sets are statistically significant (p-value < 0.01).

9.6.3 Comparative Results

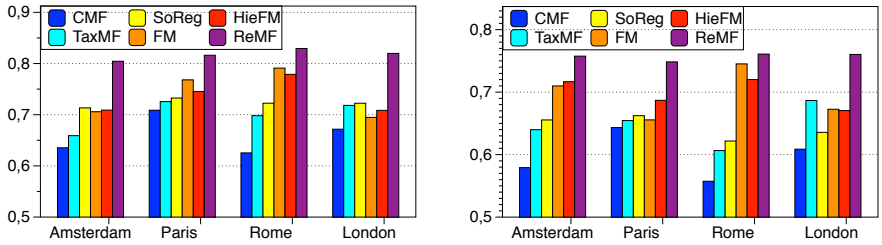
Rating Performance. Two views are created for each data set: 1) the "All" view includes all users; while 2) the "Cold start" view indicates that only users with ≤ 5 ratings are involved in the test set. Table 9.4 compares the performance of the considered recommendation methods for all data sets. Unsurprisingly, the basic matrix factorization model is consistently outperformed by feature-based recommendation methods; this shows that, in the context of the targeted evaluation scenario, the usage of auxiliary information about users positively affects recommendation accuracy. In addition, FM outperforms CMF, TaxMF and SoReg. This could be explained by FM considering item-feature interactions, in addition to user-item and user-feature interactions.

HieMF in general outperforms FM, suggesting that information about feature relationships (paths) can help predicting user preferences. `ReMF` consistently outperforms the methods in the comparison pool, with an average performance gain (w.r.t. the second best method) of **7.20%** (MAE) and **15.07%** (RMSE). Paired t-test shows that the improvements of `ReMF` on all data sets are significant (p-value < 0.01). Such big improvements clearly show the effectiveness of recursive regularization, and the advantage derived from the full inclusion of information about feature relationships.

Table 9.4 (data view "Cold start") reports the results with cold start users. As in the previous case, `ReMF` achieves the best performance compared with other methods, and significantly outperforms the second best methods in all data sets (p-value < 0.01) by **12.02%** and **17.53%** w.r.t. MAE and RMSE respectively. The relatively larger improvements on the testing view "Cold start" than on "All" indicates that `ReMF` has higher capability in coping with the cold start problem compared to the state-of-the-art methods.

Ranking Performance. We further evaluate the ranking quality of items recommended by `ReMF` and other methods in the comparison pool. Results are shown in Figures 9.6(a-b) for data sets from Instagram and Twitter, respectively. `ReMF` significantly outperforms the second best method (p-value < 0.01) on all data sets by **9.86%** on average, reaching an averaged AUC of 0.8175 in Instagram and 0.7568 in Twitter. These observations show that the influence of feature hierarchy modeled by recursive regularization can effectively complement user-item interaction data in ranking prediction.

Generalizability. We test the performance of `ReMF` on another task, i.e. product recommendation, using the data from Amazon web store [151]. Dif-



(a) AUC - Instagram (b) AUC - Twitter

Figure 9.6: AUC of `ReMF` and the comparative methods on POI data sets of four cities, through the lens of (a) Instagram and (b) Twitter.

ferent from the POI data sets, here we consider the feature hierarchy of items. We focus on the product category of "Clothing, Shoes & Jewelry", having maximal depth of 7, and an unbalanced feature hierarchy. An example path in the hierarchy from the root feature to the leaf is "Clothing, Shoes & Jewelry → Men → Accessories → Wallets". We uniformly sample the raw data set to include 100,810 ratings performed by 34,817 users to 45,716 items. Table 9.5 compares the performance of `ReMF` and the other methods in the comparison pool, measured by RMSE, which is more indicative of large errors than MAE. As in the previous setting, `ReMF` significantly outperforms the second best method (p-value < 0.01), i.e. HieFM, by **5.46%** on the testing view of "All" and **7.42%** on "Cold start". These results show that `ReMF` can be effective in multiple recommendation tasks, and with different topologies of features hierarchy.

	CMF	**TaxMF**	**SoReg**	**FM**	**HieFM**	**ReMF**
All	1.6356	1.3921	1.3912	1.3899	1.3847*	**1.3091**
Cold start	1.6386	1.4057	1.4054	1.4074	1.4033*	**1.3242**

Table 9.5: Performance (RMSE) on the testing views "All" and "Cold start" of Amazon data set. The best performance is boldfaced; the runner up is labelled with "*". All improvements by the best method are statistically significant (p-value < 0.01).

9.7 Conclusion

Hierarchies are a common way to capture relationships between features. Yet, the value of this additional information is not fully exploited by state-of-the-art feature-based recommendation methods. This chapter proposes a novel regularization method named recursive regularization for modeling the

co-influence of features in the hierarchy on user-item interactions. Based on this, a new recommendation framework `ReMF` is proposed to learn hierarchical feature influence from historical user-item interaction data. Experimental validation on real-word data sets shows that `ReMF` can largely outperform state-of-the-art methods, proving the value residing in the exploitation of feature hierarchies for better learning user and item latent factors.

We stress how recursive regularization does not only apply to tree-like data structures (hierarchy), but also to a forest of trees: adding a root feature transforms a set of trees to one tree. Generalization to graphs is less trivial, and therefore left to future work.

Chapter 10

Exploiting both Vertical and Horizontal Dimensions of Feature Hierarchy for Effective Recommendation

This chapter focuses on the relationships of features that can be induced from the horizontal dimension of a feature hierarchy. We define two types of horizontal relationships, namely complementary and alternative relationships, and show their presence in real-world datasets. We then propose a unified recommendation framework, i.e. HieVH, to seamlessly fuse both vertical and horizontal dimensions of a feature hierarchy for effective recommendation.

This chapter is published as "Exploiting both Vertical and Horizontal Dimensions of Feature Hierarchy for Effective Recommendation" [207], by Z. Sun, J. Yang, J. Zhang, and A. Bozzon in Proceedings of the 31st AAAI Conference on Artificial Intelligence, pages 189-195. AAAI, 2017.

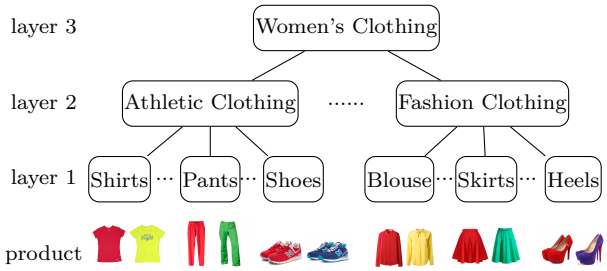


Figure 10.1: Running example of feature hierarchy.

10.1 Introduction

Feature-based recommendation has been widely studied to resolve *data sparsity* and *cold start* problems in recommender systems. Generally, features of users and items can be organized in different structures, e.g. flat or hierarchical. Feature hierarchy (FH) [95] – as a natural yet powerful structure to describe human knowledge – has proven to be effective to boost recommendation accuracy.

Early work incorporates FH for better recommendation by converting it to a flat structure [246, 123, 115, 158]. The reduction to a simpler knowledge structure, while simplifying the formalization of the recommendation problem, brings severe information loss. Recently a few studies consider the structured nature of FH, assuming that items are characterized by the affiliated features in the hierarchy. Such characterization can be modeled as the *influence* of features on user-item interactions, which is either manually defined [95], or automatically learnt from data [233].

All the methods above consider the influence of features in vertical dimension of the hierarchy, i.e. only features with child-parent affiliation have influence on user-item interactions. They ignore, however, another important dimension of FH, i.e. the horizontal dimension. Sibling and cousin features, i.e. positioned in the same layer of the hierarchy, might capture latent relationships that could be used to better characterize user-item interactions, and, consequently, to enhance recommendation accuracy. In the following we use a running example to illustrate how the horizontal dimension of FH can help characterize user-item interactions.

Running Example. Consider a Web product recommender system, where the goal is to recommend products to users. Figure 10.1 depicts a 3-layer category hierarchy of Women's Clothing. Categories in this hierarchy are organized in vertical dimension (e.g. Shirts and Athletic Clothing) or horizontal dimension (e.g. Shirts and Pants). Suppose a customer who prefers

athletic style to fashion style. She may buy more items of Athletic Clothing, such as athletic shoes and pants to match each other, instead of items of Fashion Clothing, e.g. heels or skirts. In this case, the two sibling categories Athletic Clothing and Fashion Clothing at layer 2 are characterized by an *alternative* relationship, as they are purchased by the user in a mutually exclusive fashion. The sibling categories at layer 1, Athletic Shoes and Pants, are characterized by a *complementary* relationship, as they are jointly purchased by the user. Whereas the cousin categories at this layer, e.g. Athletic Shoes and Heels are *alternative* as determined by the relationship of their parent categories.

The example highlights that feature relationships in horizontal dimension can provide additional characterization of user-item interactions. It is, however, nontrivial to exploit such kind of relationships, as the vertical affiliation of features in different layers should also be preserved. As illustrated in the above example, users' preferences on items (e.g. athletic shoes and heels) could also be affected by the relationships of their vertically affiliated features across different layers (e.g. Shoes - Athletic Clothing, Heels - Fashion Clothing). In other words, it is often impossible to disentangle the horizontal dimension from the vertical one.

Hence, this chapter contributes a unified recommendation framework HieVH that seamlessly exploits both dimensions of FH, to boost recommendation accuracy. To model the vertial dimension HieVH adapts latent factors of items by adding weighted aggregation of their affiliated features' latent factors, to better model item latent factors. The weights are automatically learnt from data. Horizontally, feature relationships are incorporated as regularizers at each layer of the hierarchy, to better model feature latent factors. In doing so, through the adaption of item latent factors with feature latent vectors in vertical dimension, feature relationships in horizontal dimension can be inherited by items. The result is a method that can seamlessly fuse vertical and horizontal dimensions of FH. While existing methods (e.g. ReMF [233]) consider vertical dimension, we stress it is nontrivial to extend them to integrate horizontal dimension, due to the lack of a matching mechanism in vertical dimension such as the use of feature latent factors.

Extensive experiments on four real-world datasets show that our approach achieves superior performance over state-of-the-art counterparts, with an average improvements of 5.23% on AUC. Besides, by uncovering the semantically rich feature relationships (*alternative* and *complementary*) between the recommended and rated items, HieVH provides better interpretations of the generated recommendations.

10.2 Related Work

Mapping Feature Hierarchy into Flat. Generic feature-based recommendation methods, including collective matrix factorization (CMF) [202, 134], factorization machine (FM) [179, 181], and SVDFeature [42], are originally designed for incorporating features organized in a flat structure. Early methods incorporating FH [246, 224] model a user's taxonomy preferences as a flat feature vector. Later, some latent factor model (LFM) based methods [199] have been designed. For example, [123, 158, 140, 115] propose adding feature latent vectors into user or item latent factors. Despite this, blending FH into all the above models requires converting the hierarchy into a flat structure, thus losing the structural information encoded in the hierarchy.

Modeling Vertical Dimension of Feature Hierarchy. Menon et al. (2011) propose an ad-click prediction method that considers FH of ads. However, it assumes that an ad is conditionally independent from all higher layer features. He et al. (2016) devise a visually-aware recommendation model by manually defining the feature influence in vertical dimension of the hierarchy. Recently, Yang et al. (2016) design a recommendation method that automatically learns such influence on user/item latent factors by a parameterized regularization traversing from root to leaf features.

These studies, however, are limited to feature influence of vertical dimension, ignoring feature relationships of horizontal dimension. Besides, they are nontrivial to be extended to seamlessly integrate horizontal feature relationships, due to the lack of a matching mechanism in vertical dimension (e.g. feature latent factors). In our unified approach, we seamlessly model both dimensions of FH, and further consider semantically rich feature relationships, i.e. *alternative* and *complementary* [149, 150].

Modeling Implicit User/Item Hierarchy. Recently, Zhang et al. (2014) and Wang et al. (2015) propose to model implicit hierarchical structure within users and/or items, based on historical user-item interactions. Our work differs from these two models, in that we consider leveraging *explicit* FH to guide the learning of latent factors.

10.3 Measuring Feature Influence and Relationships

This section first introduces our metrics for measuring feature influence in vertical dimension, and feature relationships in horizontal dimension of FH. To demonstrate the need for richer feature hierarchy characterization of user-

item interactions for better recommendation, we then apply the proposed metrics to analyze Amazon Web store data.

10.3.1 Metrics for Feature Influence and Relationships

Let $\mathcal{U}, \mathcal{I}$ denote the set of users and items, and $\mathcal{F}$ denote the set of features organized in a hierarchy. r_{ui} is the rating given by user $u \in \mathcal{U}$ to item $i \in \mathcal{I}$. Each item $i \in \mathcal{I}$ is affiliated with a subset of features $\mathcal{F}(i) = \{f_i^1, f_i^2, \ldots, f_i^L\}$, organized as a path from leaf feature f_i^1 to root feature f_i^L. Let $P(e_i)$ denote the probability of the event that an item i is rated by a user, defined as,

$$P(e_i) = \frac{|\{u|u \in \mathcal{U}, r_{ui} \neq 0\}|}{|\mathcal{U}|}$$

Based on this definition, we use *item co-occurrence* IC to measure the closeness of two items.

Definition 3 (Item Co-occurrence)

$$\mathrm{IC}(i,j) = \frac{P(e_i \cap e_j)}{P(e_i) \times P(e_j)}$$

where $P(e_i \cap e_j)$ is the joint probability of the event that both items i and j are rated by a user.

The IC measure can be used to define both feature influence in vertical dimension, and feature relationships in horizontal dimension of the hierarchy, as illustrated below.

Definition 4 (Feature Influence of Vertical Dimension) *Given the items $i_1, i_2, \ldots$ characterized by a same subset of feature path $\mathcal{F}(i_1) = \mathcal{F}(i_2) = \ldots = \{f^1, \ldots, f^L\}$, the influence of an arbitrary feature f^l $(1 \leq l \leq L)$ in the path on these items is defined as the following vector,*

$$\overrightarrow{\mathrm{FI}}(f^l) = \frac{1}{|f^l| - 1}\Big[\sum_{j \in f^l, i_1 \neq j} \mathrm{IC}(i_1, j), \sum_{j \in f^l, i_2 \neq j} \mathrm{IC}(i_2, j), \ldots\Big]$$

where each element in the vector is the average IC between the target item and all the items affiliated to the feature. This definition allows us to test the difference among the influence of features in the same path.

We then define feature relationships in horizontal dimension, based on item relationships formalized as follows.

Definition 5 (Item Relationships) *Items i, j are alternative if $P(e_i|e_j)<P(e_i)$ and $P(e_j|e_i)<P(e_j)$; they are complementary if $P(e_i|e_j)>P(e_i)$ and $P(e_j|e_i)>P(e_j)$.*

Two items i and j are therefore *alternative*, if the probability of i being rated given j is rated (e.g. $P(e_i|e_j)$), is lower than that without knowing whether j is rated or not (e.g. $P(e_i)$). Contrarily, they are complementary if the former is larger.

We now turn to the quantification of item relationships, which will be used later for measuring feature relationships. It turns out that, IC can be a proper metric for measuring item relationships, according to the following theorem.

Theorem 3 (Item Relationships Measured by IC)

$$\begin{array}{lcl} \text{Items } i \text{ and } j \text{ are alternative} & \Longleftrightarrow & \text{IC} < 1 \\ \text{Items } i \text{ and } j \text{ are independent} & \Longleftrightarrow & \text{IC} = 1 \\ \text{Items } i \text{ and } j \text{ are complementary} & \Longleftrightarrow & \text{IC} > 1 \end{array}$$

Smaller values of IC (< 1) *indicate stronger alternative relationships between items i and j; vice versa, larger values of* IC (> 1) *indicate stronger complementary relationships between items i and j.*

Proof: Using the relationship between joint probability and conditional probability, $P(e_i \cap e_j)=P(e_j|e_i)\times P(e_i)$, we have

$$\text{IC}(i,j) = \frac{P(e_i \cap e_j)}{P(e_i) \times P(e_j)} = \frac{P(e_j|e_i) \times P(e_i)}{P(e_i) \times P(e_j)} = \frac{P(e_j|e_i)}{P(e_j)}$$

Similarly, with $P(e_i \cap e_j)=P(e_i|e_j)\times P(e_j)$, we have $\text{IC}(i,j)=\frac{P(e_i|e_j)}{P(e_i)}$. Thus, we can see that if $\text{IC}(i,j)<1$, then $P(e_j|e_i)<P(e_j)$ and $P(e_i|e_j)<P(e_i)$, vise versa, suggesting an *alternative* relationship between items i and j is equivalent to $\text{IC}(i,j) < 1$. A smaller value of IC would indicate a larger gap between $P(e_j|e_i)$ and $P(e_j)$, $P(e_i|e_j)$ and $P(e_i)$, i.e. a stronger *alternative* relationship; the opposite also holds, i.e. a stronger *alternative* relationship indicates a smaller value of IC. If $\text{IC}(i,j)>1$, then $P(e_j|e_i)>P(e_j)$ and $P(e_i|e_j)>P(e_i)$, vice versa, suggesting a *complementary* relationship between items i and j is equivalent to $\text{IC}(i,j) > 1$. A larger IC indicates a stronger *complementary* relationship;

the opposite also holds. Similarly, if $\mathrm{IC}(i,j)=1$, then $P(e_j|e_i)=P(e_j)$ and $P(e_i|e_j)$=$P(e_i)$, vice versa, hence items i,j are *independent* and $\mathrm{IC}(i,j)$=1 are equivalent. □

The independence between two items provides no additional characterization of user preferences, thus it is neither beneficial for recommendation. We do not consider it particularly useful before; it will be, however, properly handled by our proposed method. With the above metric for measuring item relationships, we now propose the metric for feature relationships in horizontal dimension.

Definition 6 (Feature Relationships in Horizontal Dimension) *The relationship of two features f and g is given by*

$$\mathrm{FR}(f,g)=\frac{1}{|f|\times|g|}\sum\nolimits_{i\in f}\sum\nolimits_{j\in g}\mathrm{IC}(i,j)$$

FR is defined as the average IC between all pairs of items, where the two items in each pair are characterized respectively by the two features. Similar to IC, $\mathrm{FR}(f,g)<1,>1,=1$ indicate that features f and g are *alternative*, *complementary* and *independent*, respectively.

10.3.2 Feature Influence and Relationships in Real-world Data

We now show the presence of feature influence and relationships in the Amazon Web store data – Clothing, Shoes & Jewelry.[2] For demonstration purpose, we only show the results of the top-3 layers of the hierarchy.

The hierarchy contains 116 categories at Layer 1, thus 116 paths in vertical dimension. Comparing the influence of features in the same path, we find that 74.33% of feature influence at Layer 1 is significantly larger than that at Layer 2, and that 72.57% of the influence at Layer 2 is significantly larger than that at Layer 3 (Paired t-test, p-value < 0.01).

The root layer (layer 3) contains two features, Women's Clothing and Men's Clothing. It can be observed from Figure 10.2 that the two features have an *alternative* relationship, indicating that women and men clothing are generally not purchased together. For layer 2, we observe that the feature

[2]The detailed information about the dataset is deferred to the Experimental Results section.

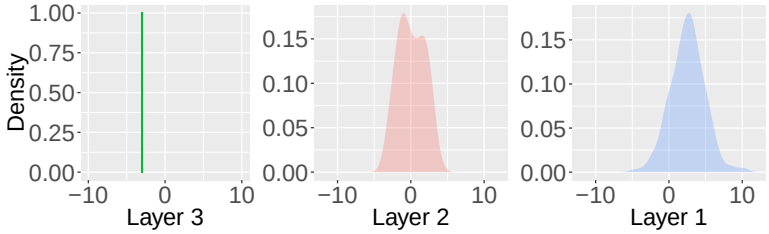


Figure 10.2: Distribution of feature relationships (log-scaled, i.e. $x = log(\text{FR})$ at 3 layers of the hierarchy.

relationships are evenly distributed on the side $log(\text{FR}) < 0$ and $log(\text{FR}) > 0$, indicating that both *alternative* and *complementary* feature relationships exist at this layer. An example of *complementary* features is Woman Watches, Man Watches, suggesting that women and men watches are usually purchased together, e.g. for couples, despite of the fact that Women's clothing and Men's clothing are *alternative*. When looking at feature relationships at layer 1, we can see that the relationships among most features are *complementary*, e.g. Women active clothing and Women athletic shoes. Overall, as a general trend, more *complementary* relationships can be observed in lower layers than upper layers, suggesting that customers tend to buy items characterized differently by fine-grained features to match each other.

10.4 The HieVH Framework

This section describes the HieVH framework – that seamlessly exploits both vertical and horizontal dimensions of FH to boost recommendation performance.

10.4.1 The Basic Recommendation Model.

Our approach is built on the latent factor model (LFM), where each user and item in the high dimensional user-item interaction space are mapped into a low-rank space. We generalize the basic LFM to seamlessly incorporate both vertical and horizontal dimensions of FH by minimizing the following equation:

$$\mathcal{J} = \overbrace{\sum_{o_{ui} \neq 0} C\big(r_{ui}, <\theta_u, \overline{\theta}_i>\big)}^{\text{cost function}} + \overbrace{\alpha \sum_{f,g \in \mathcal{F}} \Psi(\theta_f, \theta_g) + \Omega(\Theta)}^{\text{regularizers}}$$

where $o_{ui} = 1$ if user u rates item i, otherwise 0; $\theta_u, \theta_i, \theta_f \in \mathbb{R}^d$ are the latent factors of user u, item i and feature f, respectively; d is the dimension of latent factors; $C(\cdot)$ is a convex cost function (e.g. quadratic function) measuring the difference between the real rating r_{ui} and the predicted rating, i.e., the inner product of θ_u and $\overline{\theta}_i$; and $\overline{\theta}_i = \Phi(\theta_i, \theta_f)$ is the adaptive item latent factor considering the influence of features in vertical dimension on item latent factors through function Φ; Ψ is the regularization function to constrain the difference between θ_f and θ_g based on the relationships among features in horizontal dimension; α controls the importance of Ψ; $\Omega(\Theta)$ with $\Theta = \{\lambda, \theta_u, \theta_i, \theta_f\}$ are regularizers to avoid over-fitting; λ is the regularization hyperparameter. The main challenge is how to effectively formulate the functions Φ, Ψ by integrating the influence and relationships of features in the two dimensions of FH.

10.4.2 Modeling Vertical Dimension

Features are vertically affiliated in the hierarchy. Based on the results shown in the previous section, we observe that an item i is characterized by all the affiliated features $\mathcal{F}(i) = \{f_i^1, f_i^2, \ldots, f_i^L\}$, organized as a path in the hierarchy with different degrees. Hence, we formulate the function $\Phi(\theta_i, \theta_f)$ to adapt the latent factor of item i, i.e., θ_i, by adding to it the latent factors of its affiliated features, i.e., $\mathcal{F}(i)$ in the hierarchy, given by:

$$\overline{\theta}_i = \Phi(\theta_i, \theta_f, \vartheta_f) = \theta_i + \left[\vartheta_{f^1}, \vartheta_{f^2}, \cdots, \vartheta_{f^L}\right] \begin{bmatrix} -\theta_{f^1}- \\ -\theta_{f^2}- \\ \cdots \\ -\theta_{f^L}- \end{bmatrix}_{L \times d}$$

where $\vartheta_{\mathcal{F}(i)} = [\vartheta_{f^1}, \vartheta_{f^2}, \ldots, \vartheta_{f^L}]$ is the parameter vector, indicating the different influence of features in $\mathcal{F}(i)$ on item i. It can be automatically learnt by our model; θ_{f^l} $(1 \leq l \leq L)$ is the latent vector of feature $f^l \in \mathcal{F}(i)$.

In this equation, any items, e.g. i and j, that belong to the same feature set, i.e., $\mathcal{F}(i) = \mathcal{F}(j)$, share the same parameter vector, i.e., $\vartheta_{\mathcal{F}(i)} = \vartheta_{\mathcal{F}(j)} = [\vartheta_{f^1}, \vartheta_{f^2}, \ldots, \vartheta_{f^L}]$. That is to say, the features organized in a same path influence all items belonging to the leaf feature in that path. In this way, we reduce the number of parameters and avoid over-fitting. The number of parameter vectors is the total number of the unduplicated feature paths in FH, which is equal to the size of leaf feature set. Note that, in the adap-

tive function Φ, a good estimation of feature latent factors is essential to accurately adapt item latent factors, which can be facilitated by considering horizontal dimension of FH, as given below.

10.4.3 Modeling Horizontal Dimension

From the perspective of horizontal dimension, features are organized as siblings or cousins at the same layer of the hierarchy. The analysis on real-world data shows the presence of two types of feature relationships, i.e., *alternative* and *complementary*, which are highly useful to better model feature latent factors.

Hence, we incorporate such kind of feature relationships by assuming that in each layer l $(1 \leq l \leq L)$ of the hierarchy: if two features are *alternative*, then the distance of their latent factors should be large; if *complementary*, the distance of their latent factors should be small. Based on the above assumption, we devise the following regularizer to better model feature latent factors,

$$\Psi(\theta_f, \theta_g) = \sum\nolimits_{l=1}^{L} \sum\nolimits_{f,g \in \mathcal{F}^l, f<g} \sigma_{fg} \|\theta_f - \theta_g\|_F^2$$

where $\mathcal{F}^l$ is the feature set at layer l of the hierarchy; $\sigma_{fg} = log(\mathrm{FR}(f, g))$ with $\sigma_{fg} < 0, > 0, = 0$ indicating f, g are *alternative*, *complementary* and *independent*, respectively. Through adaptive function Φ adding latent vectors of affiliated features to their items' latent factors, feature relationships in horizontal dimension are inherited by items. Consequently, the better estimated feature latent factors can more accurately adapt item latent factors.

Similarly, we also incorporate item relationships to help better model item latent factors by assuming that if two items are *alternative*, the distance of their latent factors should be large; if *complementary*, it should be small. Based on this, we design the following regularizer,

$$\Psi(\theta_f, \theta_g; \theta_i, \theta_j) = \Psi(\theta_f, \theta_g) + \sum\nolimits_{i,j \in \mathcal{I}, i<j} \sigma_{ij} \|\theta_i - \theta_j\|_F^2$$

where σ_{ij}=$log(\mathrm{IC}(i, j))$ with $\sigma_{ij}< 0, > 0, = 0$ indicating items i, j are alternative, complementary and independent.

Note that σ_{fg}, σ_{ij} seamlessly accommodate our assumptions illustrated below: if two features f, g are alternative, then we have $\mathrm{FR} < 1$, thus $\sigma_{fg} < 0$.

In this case, minimizing Ψ leads to large distance between θ_f and θ_g; if f, g are complementary, then we have FR > 1, thus $\sigma_{fg} > 0$. In this case, minimizing Ψ leads to small distance between θ_f and θ_g; if f, g are independent, then FR $= 1$, thus $\sigma_{fg} = 0$. In this case, the independent feature relationships are not considered in Ψ. σ_{ij} holds similar properties as σ_{fg}.

Once the feature and item relationships are incorporated into the objective function $\mathcal{J}$, we can more accurately model feature and item latent factors in function Φ, thus can ultimately better model user-item interactions.

Remark. HieVH seamlessly integrates the modeling of both vertical and horizontal dimensions of FH. Though in this chapter we focus on item FH, HieVH can as well accommodate user FH. It is noteworthy to remark how HieVH is able to handle arbitrarily imbalanced FH, thus making its application suitable to a wide variety of application scenarios.

10.4.4 Model Learning

We adopt the widely used stochastic gradient descent method to optimize HieVH. The update rules of all the variables are given by the following equations. The optimization process is shown in Algorithm 3, which is mainly composed of parameter update (line 3-12).

$$
\begin{cases}
\nabla \mathcal{J}(\theta_u) = \sum_{i \in \mathcal{I}} o_{ui} \left(<\theta_u, \overline{\theta}_i> - r_{ui} \right) \overline{\theta}_i + \lambda \theta_u \\
\nabla \mathcal{J}(\theta_i) = \sum\limits_{u \in \mathcal{U}} o_{ui} \left(<\theta_u, \overline{\theta}_i> - r_{ui} \right) \theta_u + \lambda \theta_i + \alpha \sum\limits_{i,j \in \mathcal{I}, i<j} \sigma_{ij} (\theta_i - \theta_j)
\end{cases}
$$

$$
\forall f \in \mathcal{F}^l, l = \{1, 2, \cdots, L\},
$$

$$
\begin{cases}
\nabla \mathcal{J}(\theta_f) = \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}, f \in \mathcal{F}(i)} \vartheta_f o_{ui} \left(<\theta_u, \overline{\theta}_i> - r_{ui} \right) \theta_u \\
\qquad\qquad + \lambda \theta_f + \alpha \sum_{f,g \in \mathcal{F}^l, f<g} \sigma_{fg} (\theta_f - \theta_g) \\
\nabla \mathcal{J}(\vartheta_f) = \sum\limits_{u \in \mathcal{U}} \sum\limits_{i \in \mathcal{I}, f \in \mathcal{F}(i)} o_{ui} \left(<\theta_u, \overline{\theta}_i> - r_{ui} \right) <\theta_u, \theta_f> + \lambda \vartheta_f
\end{cases}
$$

Complexity Analysis. The computational time is mainly taken by evaluating the objective function $\mathcal{J}$ and updating the relevant variables. The time to compute $\mathcal{J}$ is $\mathcal{O}(d|\mathbf{R}| + dn^2)$, where $|\mathbf{R}|$ is the number of non-zero observations in the rating matrix $\mathbf{R}$, and n is the number of items. For the gradients $\nabla \mathcal{J}(\theta_u), \nabla \mathcal{J}(\theta_i), \nabla \mathcal{J}(\theta_f), \nabla \mathcal{J}(\vartheta_f)$, the computational time are $\mathcal{O}(d|\mathbf{R}|)$, $\mathcal{O}\left(d|\mathbf{R}| + d\frac{n(n-1)}{2}\right)$, $\mathcal{O}\left(d|\mathcal{F}||\overline{\mathbf{R}}| + d\sum_{l=1}^{L} \frac{|\overline{\mathcal{F}}|(|\overline{\mathcal{F}}|-1)}{2}\right)$, $\mathcal{O}(L|\mathcal{F}^1||\overline{\mathbf{R}}|)$, respectively. Wherein $|\mathcal{F}|$ is the total number of features in the hierarchy; $|\overline{\mathbf{R}}|$ is the average number of ratings under each feature; $|\overline{\mathcal{F}}|$ is the average number of features at each layer of the hierarchy. Generally due to $L < |\overline{\mathcal{F}}| < |\mathcal{F}^1| <$

Algorithm 3: HieVH Optimization Process

Input: rating matrix $\mathbf{R}$, feature hierarchy $\mathcal{F}$, $d, \alpha, \lambda, \gamma, Iter$

Initialize θ, ϑ with small values;
$L \leftarrow$ the highest layer of $\mathcal{F}$;
// Parameter update for $\mathcal{J}$
for $t = 1; t \leq Iter; t++$ **do**
  foreach $u \in \mathcal{U}, i \in \mathcal{I}$ **do**
    $\theta_u^{(t)} \leftarrow \theta_u^{(t-1)} - \gamma \nabla \mathcal{J}(\theta_u)$;
    $\theta_i^{(t)} \leftarrow \theta_i^{(t-1)} - \gamma \nabla \mathcal{J}(\theta_i)$;
    for $l = 1; l \leq L; l++$ **do**
      foreach $f \in \{\mathcal{F}^l \cap \mathcal{F}(i)\}$ **do**
        $\theta_f^{(t)} \leftarrow \theta_f^{(t-1)} - \gamma \nabla \mathcal{J}(\theta_f)$;
        $\vartheta_f^{(t)} \leftarrow \vartheta_f^{(t-1)} - \gamma \nabla \mathcal{J}(\vartheta_f)$;
  if $\mathcal{J}$ *has converged* **then**
    break;

$|\mathcal{F}| \ll n$ and $|\overline{\mathbf{R}}| < |\mathbf{R}|$, the overall computational complexity of Algorithm 3 is $(Iter \times \mathcal{O}(d|\mathbf{R}| + dn^2))$. In summary, our proposed HieVH framework is scalable to large datasets.

10.5 Experimental Results

10.5.1 Experimental Setup

Datasets. To validate HieVH, we use the Amazon Web store dataset [151]. This dataset has recently been applied for evaluating recommendation methods incorporating FH [95, 233]. Similar to these work, we consider the Clothing Shoes & Jewelry dataset; to evaluate the generalizability of HieVH, we further consider three other datasets in different domains, including Electronics, CDs & Vinyl, and Home & Kitchen. The FHs of all the datasets are imbalanced. We uniformly sample the datasets to balance their sizes for cross-dataset comparison. Table 10.1 reports the statistics of the datasets.

Datasets	#users	#items	#ratings	#features	#layers
Clothing.	36,000	42,201	60,141	2,764	7
Electronics	43,234	38,766	77,962	1,292	6
CDs & Vinyl	33,868	36,320	71,872	1,293	6
Home & Kitchen	44,519	37,445	73,820	2,002	5

Table 10.1: Descriptive statistics of the datasets.

Evaluation. Standard 5-fold cross validation is adopted to evaluate our proposed model. The Area Under the ROC Curve (AUC) is used as the evaluation metric. Larger AUC indicates better recommendation performance.

Comparison Methods. We compare with six state-of-the-art algorithms, 1) **MF** [192]: matrix factorization model; 2) **CMF** [202]: collective MF; 3) **FM** [179]: factorization machine; 4) **TaxMF** [42, 123]: taxonomy based MF ; 5) **Sherlock** [95]: visually-aware model ; 6) **ReMF** [233]: recursive regularization based MF. Methods 2-3 only utilize features in FH without considering the structure. Methods 4-6 are all based on FH. Besides, three variants of our proposed framework are compared. A) **HieV**: only considers vertical feature influence; B) **HieVC**: exploits vertical feature influence and horizontal *complementary* feature relationship; C) **HieVH**: fuses both vertical feature influence and horizontal *complementary* & *alternative* feature relationships.

Parameter Settings. Optimal parameter settings have been empirically estimated. We set $d = 10$ and apply a grid search in $\{0.001, 0.01, 0.1\}$ for γ, λ and 1/2-way regularization of FM; $\alpha = 0.5, 0.01$ for CMF and ReMF, respectively; for Sherlock, we use the same settings as [95].

10.5.2 Impact of α

In HieVH, α controls the importance of feature relationships in the horizontal dimension of FH. We apply grid search in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ to investigate the impact of α on recommendation performance. Results are shown in Figure 10.3. As α varies from small to large, the performance first increases then decreases, with the maximum reached at the range $[10^{-3}, 10^{-2}]$. The performance variations across datasets suggest the need for dataset-specific settings; the similarity in performance variation across α values demonstrates the robustness of HieVH.

10.5.3 Comparative Results

Table 11.2 summarizes the performance of all comparison methods across all datasets, where two views are created for each dataset: 'Warm Start' indicates all users are considered in the test data; while 'Cold Start' indicates only users with ≤ 5 ratings are involved in the test data. Several interesting observations can be noted.

Datasets	Cases	MF	CMF	FM	TaxMF	Sherlock	ReMF	HieV	HieVC	HieVH	Improve
Cloth, Shoes & Jewelry	Warm Start	0.5455	0.5646	0.6826	0.6509	0.6747	0.7015*	0.7160	0.7291	**0.7375**	5.13%
	Cold Start	0.5426	0.5667	0.6629	0.6493	0.6702	0.7032*	0.7124	0.7284	**0.7352**	4.55%
Electronics	Warm Start	0.5555	0.5762	0.6839	0.6569	0.6915	0.7337*	0.7512	0.7672	**0.7748**	5.60%
	Cold Start	0.5526	0.5735	0.6831	0.6475	0.6982	0.7305*	0.7474	0.7658	**0.7741**	5.97%
CDs & Vinyl	Warm Start	0.5478	0.5622	0.6356	0.6905	0.7082	0.7249*	0.7328	0.7516	**0.7600**	4.84%
	Cold Start	0.5433	0.5609	0.6231	0.6881	0.7076	0.7243*	0.7315	0.7514	**0.7588**	4.76%
Home & Kitchen	Warm Start	0.5420	0.5545	0.6938	0.6469	0.6938	0.7279*	0.7456	0.7574	**0.7667**	5.33%
	Cold Start	0.5395	0.5562	0.6915	0.6511	0.6973	0.7275*	0.7412	0.7554	**0.7650**	5.15%

Table 10.2: Performance (AUC) of comparison methods. The best performance is highlighted in bold; the second best performance of other methods is marked by '*'; 'Improve' indicates the relative improvements that HieVH achieves w.r.t. the '*' results.

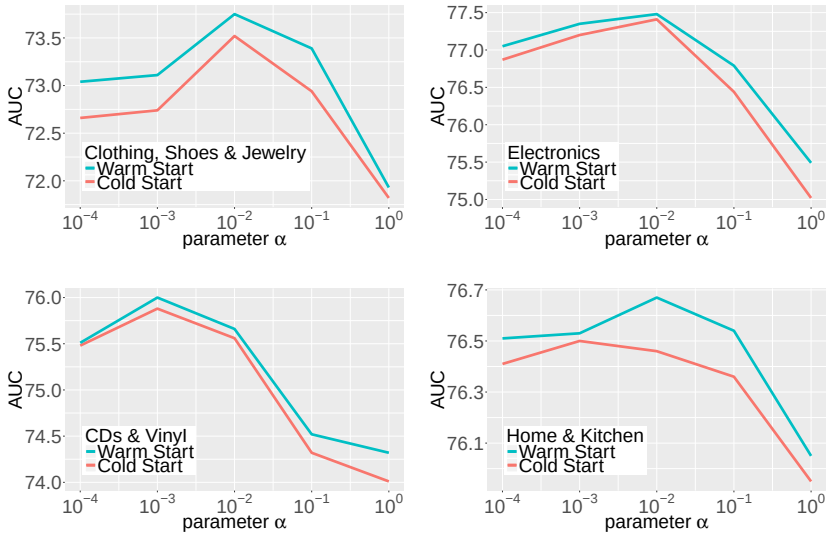


Figure 10.3: The impact of parameter α.

Method	Layer 1		Layer 2	
	Cp	*Ap*	*Cp*	*Ap*
ReMF	87.62%	12.38%	75.23%	24.76%
HieV	88.89%	11.11%	76.19%	23.89%
HieVC	92.62%	7.38%	84.62%	15.38%
HieVH	95.65%	4.35%	89.35%	10.65%

Table 10.3: *Cp* and *Ap* of the Clothing hierarchy.

Compared with all other methods incorporating FH, MF considering no auxiliary information performs the worst, indicating the effectiveness of feature based recommendation. The methods originally designed for the flat feature structure, including CMF and FM, generally perform worse than the FH based methods (TaxMF, Sherlock and ReMF). Since FH needs to be converted into a flat structure when applied into CMF and FM, the result demonstrates that useful information is lost in the conversion. FM outperforms CMF and even some FH based methods. This could be explained by FM further considering user-feature interactions, in addition to the user-item and item-feature interactions, as in CMF.

Among the three state-of-the-art FH based methods, Sherlock performs better than TaxMF, but worse than ReMF. The reason behind is that TaxMF views the influence of features in different layers of FH identically, whereas Sherlock weights the influence of features in different layers differently. However, the weights are defined manually. In contrast, ReMF automatically

learns such influence by a parameterized regularization traversing from root to leaf features.

We now compare the three variants of our framework – HieV, HieVC and HieVH, with the recently proposed ReMF. By considering vertical feature influence only, HieV performs slightly better than ReMF. The possible explanation is that in HieV, item latent factors are directly adapted by the affiliated feature latent factors; whereas in ReMF, latent factors of items are regularized by those of items that share common ancestor features, which means items are indirectly influenced by their affiliated features. In other words, the adaption of item latent factors in HieV is more straightforward than that in ReMF, thus more effective. HieVC upgrades HieV by adding *complementary* feature relationships in the horizontal dimension; HieVC is then promoted to HieVH by further incorporating *alternative* feature relationships. In results, HieVC performs better than HieV, but worse than HieVH, implying that both *complementary* and *alternative* feature relationships among horizontally organized features help improve recommendation accuracy.

Overall, when comparing with all the other comparison methods across all the datasets, HieVH achieves the best performance. The improvements w.r.t. Warm Start and Cold Start are 5.23%, 5.11% on average, respectively, which are statistically significant (Paired t-test, p-value < 0.001). This implies that the recommendation performance can be further enhanced by appropriately considering both vertical and horizontal dimensions of FH.

Interpretations by HieVH. We now analyze how the incorporation of feature relationships can better explain user-item interactions. To this end, we first derive for each user the feature relationships between the rated items (i.e. training data), and the correctly recommended items (i.e. intersection between recommended items and test data). We calculate the percentage of complementarity Cp and alternativity Ap among these relationships for each user. Good recommendations would result in a high percentage of complementary items and a low percentage of alternative items.

Table 10.3 shows the average Cp and Ap for all users in the test data at the top-3 layers of the Clothing hierarchy (layer 3 excluded since only an alternative relationship exists). We could see that from ReMF, HieV to HieVC, HieVH, with *complementary* and *alternative* feature relationships considered, Cp increases, and Ap decreases in both layers. Among all the methods, HieVH achieves the highest Cp, and lowest Ap, with significant improvements over HieVC (Paired t-test, p-value < 0.01). This clearly indicates that by incorporating the two types of feature relationships, the recommen-

Figure 10.4: The example of recommendation.

dations better approximate real user preferences. Example users to whom the recommendations benefit from features relationships generated by HieVH are shown in Figure 10.4. The recommendations to users u_1 and u_2 are better because of the *complementarity* among fashion clothing that u_1 is more fond of, and among athletic clothing that u_2 instead is more fond of, and the *alternativity* between fashion and athletic clothing. Similarly, the recommendations for u_3 are provided because of her interests in fashion clothing and lingerie. For user u_4, it is discovered that he likes clothing collocation, i.e. the *complementarity* of the items he purchased.

10.6 Conclusion

Feature hierarchy is known to enhance recommendation performance. Existing methods only consider feature influence in vertical dimension, ignoring feature relationships in horizontal dimension. In this chapter, we first show the presence of feature influence and relationships in real-world datasets based on our proposed metrics. Then we design the HieVH to seamlessly exploit both the vertical and horizontal dimensions of feature hierarchy for better recommendation. Experimental results on four real-world datasets show that HieVH consistently outperforms state-of-the-art methods. Besides, HieVH provides better interpretations of the generated recommendations.

Chapter 11

MRLR: Multi-level Representation Learning for Personalized Ranking in Recommendation

This chapter explores the effectiveness of state-of-the-art neural network based methods for learning representations of users and items in recommendation. We design a unified Bayesian framework MRLR to adapt the general representation learning for recommendation, meanwhile to learn high-quality user and item representations from both user-item interactions and a multi-level item organization.

This chapter is published as "Exploiting both Vertical and Horizontal Dimensions of Feature Hierarchy for Effective Recommendation" [208], by Z. Sun*, J. Yang*, J. Zhang, A. Bozzon, C. Xu, and Y. Chen in Proceedings of the 26th International Joint Conference on Artificial Intelligence (to appear). 2017. (*: Joint first authors.)

11.1 Introduction

Recommendation is a fundamental task on the Web to mitigate the information overload problem [206]. Recently, representation learning (RL) has attracted a considerable amount of interest from various domains, with recommender systems being no exception [83, 133, 215, 52, 16]. The popularization of RL in recommendation can be mainly attributed to the word embedding techniques (e.g. CBOW and Skip-gram [155, 156]) originated from the natural language processing (NLP) domain. Word embedding generally refers to the low-dimensional distributed representation of words [17], capturing syntactical and semantic relationships among words. The fast development of RL has enabled a series of methods for NLP tasks, among which the most significant are the extensions of word embedding to learn textual representations in different levels of granularity (e.g. document or paragraph RL [128]), so as to help capture richer relationships between words and paragraphs or documents.

In recommendation, the RL method is applied to capture local relationships between items, thus called item embedding. Item embedding learns low-dimensional item representation by modeling item co-occurrence in individual user's interaction record, thus enhancing recommendation performance. While this helps to learn better item representation, item embedding alone (e.g. Item2Vec [16], CoFactor [133], Meta-Prod2Vec [215]) does not allow for personalized recommendation. Inspired by document RL (e.g. PV-DM [128]), an important branch of work explores the potential of item embedding in personalized recommendation by learning representations for both users and items – as documents and words respectively in NLP (e.g. User2Vec [83]).

We argue that the potential of RL for recommendation has not been fully exploited. Two major aspects have been largely neglected. First, recommendation is essentially a personalized ranking problem, while existing RL methods only model recommendation as a rating prediction problem. Second, existing methods all ignore the possible multi-level organizations of items for uncovering fine-grained item relationships in recommendation (similar as word-paragraph-document in NLP), which could in turn help achieve better personalized ranking performance.

Personalized Ranking. It has been shown that recommendation is better modeled as a personalized ranking problem [223, 180, 197]. Existing RL methods, however, optimize towards predicting user preferences over individual items (i.e. rating prediction), instead of predicting user preferences over a list of items (i.e. personalized ranking).

We therefore advocate for a RL method specifically designed for personalized ranking. It is however non-trivial to adapt item embedding to personalized ranking. The original item embedding method only learns from item co-occurrence relationships, whereas for personalized ranking the method has to learn from user-specific lists of items ranked in terms of user preferences. We hence first extend the original embedding method to a more generic Bayesian framework, under which we then fuse in the likelihood function of user-specific pairwise item ranking. This unified framework can then learn user and item embedding from both item co-occurrence relationships and user-specific ranked lists of items, benefiting from user and item RL while reaching the goal of personalized ranking based recommendation.

Multi-level RL. To fully exploit RL for better recommendation, we further extend our proposed personalized ranking framework to multi-level RL, so as to capture fine-grained item relationships. Our method is inspired by paragraph in NLP as the intermediate level of word organization between individual words and documents. Intuitively, each paragraph conveys a main message, and all the words in the paragraph should help support such message. In analogy to paragraph, we introduce item category as the intermediate level of item organization between individual item and items rated by the same user, since items with the same category often share similar characteristics. For example, online products are often described by categories as metadata such as clothing, books, electronics, and so on.

Our unified Bayesian framework therefore facilitates multi-level RL by combining RL in all the three levels (i.e. individual item, item category, and user). While item category has recently been intensively studied [95, 233], we are the first to investigate it from the perspective of multi-level RL, which enables our framework to capture the relationships of items in local context (i.e. item co-occurrence relationships), in the same category, and in user-specific ranked item list.

Original Contributions. In summary, we contribute a multi-level representation learning method for personalized ranking based recommendation (MRLR). To the best of our knowledge, we are the first to adopt RL for personalized ranking; meanwhile, we design multi-level RL to capture fine-grained item relationships by leveraging category RL as the intermediate level RL between item RL and user RL, thus to further enhance recommendation performance. Extensive validation on multiple real-world datasets shows that MRLR can consistently outperform state-of-the-art methods, resulting in a 5.18% lift in AUC.

11.2 Related Work

Rating Prediction vs. Personalized Ranking. Recommendation is typically formulated as either a rating prediction problem or a personalized ranking one [223, 205]. Personalized ranking has proven to be more direct and efficient than rating prediction, as most recommendations in real-world scenarios are presented in a ranked item list. In general, the rating prediction based algorithms estimate user preferences towards individual items as absolute scores, based on which items are ordered and recommended to users in a ranked list. Typical methods include probabilistic matrix factorization (PMF) [159], tensor factorization (TF) [116] and factorization machine (FM) [179]. In contrast, the ranking based algorithms directly optimize towards learning users' preferences as personalized ranking on a set of items. Typical methods include CofiRank [223], Bayesian Personalized Ranking (BPR) [180], CLiMF [198].

Latent Factor Model vs. Representation Learning. Until recently, state-of-the-art algorithms for recommendation are dominated by the latent factor model (LFM) [199], which maps the high-dimensional user-item interaction data to low-dimensional latent user and item space. LFM based methods include all the representative rating prediction and ranking based methods mentioned above; in addition, many other effective recommendation methods fall into this category, such as NMF [130], CMF [202], SVDFeature [42] and SVD++ [124]. While these methods leverage global statistical information of user-item interaction data, they cannot capture fine-grained regularities in the latent factors [171].

Recently, representation learning (RL) based methods have drawn much attention. In contrast to LFM based methods, RL based methods have shown to be highly effective in capturing local item relationships by modeling item co-occurrence in individual user's interaction record [83, 16, 133]. These methods are mostly inspired by the word embedding techniques, which can be traced back to the classical neural network language model [17], and the recent breakthrough of Word2Vec techniques including CBOW and Skip-gram [155, 156].

Representation Learning in Recommendation.

Several RL based methods have been proposed to date. For example, Barkan and Koenigstein 2016 propose a neural item embedding model (Item2Vec) for collaborative filtering, which is capable of inferring item-to-item relationships. Vasile et al. 2016 extend Item2Vec to a more generic model by utilizing

side-information to help compute the low-dimensional embeddings of items. However, they all fail to provide personalized recommendation, as embedding techniques are only used to learn better item representation. Several studies extend RL for personalization. Grbovic et al. 2015 first introduce the User2Vec model, which simultaneously learns representations of items and users by considering the user as a "global context". Liang et al. 2016 propose the CoFactor model, which jointly decomposes the user-item interaction matrix and the item-item co-occurrence matrix – equivalent to item embedding [131] – with shared item latent factors. However, all these methods model recommendation as a rating prediction problem.

In contrast, we propose a RL based method by formulating recommendation as personalized ranking. Furthermore, we consider multi-level RL, which can capture fine-grained item relationships in multi-level item organizations, to fully exploit RL for better personalized ranking performance.

11.3 The Proposed MRLR Framework

In this section, we first formalize recommendation as a personalized ranking problem, and then present our multi-level RL framework (MRLR) to achieve the goal of personalized ranking. Finally, the model learning is introduced, including parameter estimation and complexity analysis.

11.3.1 Problem Formulation and Objective Function

Suppose there are m users $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$, and n items $\mathcal{I} = \{v_1, v_2, \ldots, v_n\}$. We use the binary user feedback matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$, i.e., if the interaction (rating) from u_p to v_i is observed, indicating u_p prefers v_i, then $\mathbf{R}_{pi} = 1$; otherwise, $\mathbf{R}_{pi} = 0$. $\mathcal{I}^+_{u_p}$ is the set of items that user u_p prefers. $\mathcal{D}_r = \{(u_p, v_i, v_j) | u_p \in \mathcal{U}, v_i \in \mathcal{I}^+_{u_p}, v_j \in \mathcal{I} \backslash \mathcal{I}^+_{u_p}\}$ is the set of user-specific ranking triples indicating u_p prefers v_i to v_j, where $\mathcal{I} \backslash \mathcal{I}^+_{u_p}$ denotes the set of items that u_p has no interaction with. $\mathcal{D}_c = \{(u_p, v_i, v_k) | u_p \in \mathcal{U}, v_i, v_k \in \mathcal{I}^+_{u_p}\}$ is the set of item co-rated triples indicating u_p prefers both v_i and v_k. For each user, we aim to provide a personalized ranking list of items that she has not interacted with. More specifically, our aim is to design a unified multi-level RL framework (MRLR) to learn user and item embeddings from both item co-rated relationships and user-specific ranked lists of items, thus to benefit from user and item RL, as well as to reach the goal of personalized ranking.

We define the objective function of MRLR using a Bayesian framework. To consider both item co-rated relationships and personalized ranking, our MRLR framework maximizes the following posterior probability,

$$P(\Theta|\mathcal{D}) \propto P(\mathcal{D}|\Theta)P(\Theta) \propto P(\mathcal{D}_c, \mathcal{D}_r|\Theta)P(\Theta)$$

where Θ is the set of parameters in MRLR, $\mathcal{D}$ is the observed data. It is proportional to maximizing the likelihood of the observed triples given the embeddings, i.e., $P(\mathcal{D}|\Theta)$. We define the likelihood function as the joint probability of item co-rated triples and user-specific ranking triples, i.e., $P(\mathcal{D}_c, \mathcal{D}_r|\Theta)$. Assuming the item co-rated triples and user-specific ranking triples are conditionally independent, the joint probability is then reformulated as follows:

$$\begin{aligned} &P(\mathcal{D}_c, \mathcal{D}_r|\Theta) = P(\mathcal{D}_c|\Theta)P(\mathcal{D}_r|\Theta) \\ &= \prod_{(u_p,v_i,v_k)\in\mathcal{D}_c} P((u_p, v_i, v_k)|\Theta) \prod_{(u_p,v_i,v_j)\in\mathcal{D}_r} P((u_p, v_i, v_j)|\Theta) \end{aligned}$$

where $P((u_p, v_i, v_k)|\Theta)$ denotes the conditional probability of item co-rated triples (u_p, v_i, v_k), and $P((u_p, v_i, v_j)|\Theta)$ denotes the conditional probability of user-specific ranking triples (u_p, v_i, v_j). Hence, our MRLR framework seamlessly fuses the two components: (1) item co-rated triples for better user and item embedding; (2) user-specific ranking triples for personalized ranking. Besides, through multi-level RL, MRLR can fully exploit RL from a multi-level item organization, i.e., items in user-specific ranked list, items in a same category, and individual items, to capture fine-grained item relationships for better recommendation.

11.3.2 Modeling User and Item Embedding

For each user u_p and item $v_i \in \mathcal{I}^+_{u_p}$, the Skip-gram method [155, 156] aims at predicting the probability of item $v_k \in \mathcal{I}, (i \neq k)$ also preferred by u_p, i.e. $P(v_k|v_i)$, which is calculated by the softmax function as follows:

$$P(v_k|v_i, \Theta) = \frac{exp(\mathbf{v}_i^T \mathbf{v}'_k)}{\sum_{v_g\in\mathcal{I}} exp(\mathbf{v}_i^T \mathbf{v}'_g)} \tag{11.1}$$

where $\mathbf{v}_i, \mathbf{v}_k$ are embeddings of items v_i, v_k, respectively.

To allow for personalization, we model user u_p's preference towards item v_k by a similar softmax function:

$$P(v_k|u_p, \Theta) = \frac{exp(\mathbf{u}_p^T \mathbf{v}'_k)}{\sum_{v_g\in\mathcal{I}} exp(\mathbf{u}_p^T \mathbf{v}'_g)} \tag{11.2}$$

where $\mathbf{u}_p$ denotes the user embedding of u_p.

We now proceed to model the item co-rated triples, i.e., $P((u_p, v_i, v_k)|\Theta)$. It should properly accommodate both the item co-rated relationships (Equation 11.1), and personalization (Equation 11.2). Instead of directly optimizing $P((u_p, v_i, v_k)|\Theta)$, we optimize the conditional probability $P(v_k|(u_p, v_i), \Theta)$, $P(v_i|(u_p, v_k), \Theta)$ and $P(u_p|(v_i, v_k), \Theta)$. Since we aim at recommending items to given users, we do not need to model $P(u_p|(v_i, v_k), \Theta)$. For the conditional probability of items, we decide to take $P(v_k|(u_p, v_i)\Theta)$ for example. Inspired by document RL in NLP [128], the user embedding $\mathbf{u}_p$ and item embedding $\mathbf{v}_i$ are summed as the new condition to predict the probability of v_k rated by u_p, given by,

$$P(v_k|(u_p, v_i), \Theta) = \frac{exp(\alpha_1 \mathbf{u}_p^T \mathbf{v}'_k + \alpha_2 \mathbf{v}_i^T \mathbf{v}'_k)}{\sum_{v_g \in \mathcal{I}} exp(\alpha_1 \mathbf{u}_p^T \mathbf{v}'_g + \alpha_2 \mathbf{v}_i^T \mathbf{v}'_g)} \tag{11.3}$$

where $\alpha_1 + \alpha_2 = 1.0$; $exp(\alpha_1 \mathbf{u}_p^T \mathbf{v}'_k + \alpha_2 \mathbf{v}_i^T \mathbf{v}'_k)$ aims to take into account both the personalized aspect by the term $\mathbf{u}_p^T \mathbf{v}'_k$, and item co-rated relationships by the term $\mathbf{v}_i^T \mathbf{v}'_k$. We model $P(v_i|(u_p, v_k), \Theta)$ in a similar way.

11.3.3 Modeling Personalized Ranking

We now introduce the second component of our framework, i.e., user-specific ranking triples, to achieve the goal of personalized ranking. We model $P(\mathcal{D}_r|\Theta)$ similar as $P(\mathcal{D}_c|\Theta)$. As before, we optimize the conditional probability of $P((v_j, v_i)|u_p, \Theta)$ and $P(u_p|(v_j, v_i), \Theta)$ instead of $P((u_p, v_i, v_j)|\Theta)$. Then, only $P((v_j, v_i)|u_p, \Theta)$ needs to be considered, since our goal is to recommend items. In terms of $P((v_j, v_i)|u_p, \Theta)$, it involves a user's preference over a pair of items. Equation 11.2 models a user's preference towards a certain item, based on which we further deduce a user's preference on a pair of items. As the triple (u_p, v_i, v_j) indicates that user u_p prefers item v_i to item v_j, it means that for u_p, we should maximize the probability that v_i is preferred by u_p but v_j is not favored by u_p. We denote such probability by $P((\neg v_j, v_i)|u_p, \Theta)$, which is defined as below:

$$P((\neg v_j, v_i)|u_p, \Theta) = \frac{exp(\mathbf{u}_p^T \mathbf{v}'_i - \mathbf{u}_p^T \mathbf{v}'_j)}{\sum_{v_h, v_g \in \mathcal{I}} exp(\mathbf{u}_p^T \mathbf{v}'_h - \mathbf{u}_p^T \mathbf{v}'_g)} \tag{11.4}$$

where the term $exp(\mathbf{u}_p^T \mathbf{v}'_i - \mathbf{u}_p^T \mathbf{v}'_j)$ denotes the preference difference of user u_p towards items v_i and v_j.

11.3.4 Modeling Multi-level Item Organization

We further consider multi-level granularity of item organizations to capture fine-grained item relationships. Specifically, we introduce item category as the intermediate level granularity of item organizations, between items in the same user-specific ranked list and individual items. The rationale behind is that items in a same category share similar characteristics.

To integrate the influence of item category for better recommendation, we extend our personalized ranking framework to multi-level RL for full exploitation of RL. The item embedding is thus reformulated as follows,

$$\overline{\mathbf{v}}_i = \mathbf{v}_i + \frac{\alpha_3}{|\mathcal{C}_{v_i}|} \sum\nolimits_{c_l \in \mathcal{C}_{v_i}} \mathbf{c}_l \tag{11.5}$$

where $\mathcal{C}_{v_i}$ is the set of categories that item v_i belongs to; $|\mathcal{C}_{v_i}|$ is the size of category set $\mathcal{C}_{v_i}$; $\mathbf{c}_l$ is the embedding for item category c_l. By replacing the item embedding in Equations 11.3 and 11.4, the item category RL can adapt item embedding, serving as the intermediate level RL in our framework. MRLR can now capture fine-grained relationships of items in local context (i.e., item co-rated relationships), in the same category, and in user-specific ranked item list.

11.3.5 Model Learning

Optimizing our MRLR framework is proportional to minimizing the negative log-likelihood function, given by,

$$\min_{\Theta} \mathcal{J} = - \sum_{(u_p, v_i, v_k) \in \mathcal{D}_c} logP((u_p, v_i, v_k)|\Theta) - \sum_{(u_p, v_i, v_j) \in \mathcal{D}_r} logP((u_p, v_i, v_j)|\Theta) + \lambda_\Theta \Omega(\Theta)$$

where $\Omega(\Theta)$ is the regularizer to prevent over-fitting, and λ_Θ is the regularization coefficient. To solve the optimization problem, we apply the stochastic gradient descent (SGD) method to the objective function $\mathcal{J}$.

Approximation of Softmax Function. It is impractical to directly adopt the softmax functions $P(v_k|(u_p, v_i), \Theta)$, $P(v_i|(u_p, v_k), \Theta)$ and $P((\neg v_j, v_i)|u_p, \Theta)$ to optimize our framework, since the cost of computing the denominators of these functions is proportional to the total number of items, i.e., n, which is considerably huge in real-world applications. To accelerate the speed, we adopt negative sampling proposed in [156]. Take $P(v_k|(u_p, v_i), \Theta)$ as an ex-

Algorithm 4: The optimization of MRLR

Input: $\mathbf{R}, \mathcal{C}, \lambda_\Theta, \alpha, \gamma, d, iter$

1 Initialize $\Theta = \{\mathbf{u}, \mathbf{v}, \mathbf{c}\}$ with small values;
2 Randomly sample (u_p, v_i, v_j) for $\mathcal{D}_r$;
// Negative sampling procedure
3 **foreach** $(u_p, v_i, v_k) \in \mathcal{D}_c$, *and* $(u_p, v_i, v_j) \in \mathcal{D}_r$ **do**
4 Draw N negative instances from the distribution $P(\mathcal{D}_c^-)$;
5 Draw N negative instances from the distribution $P(\mathcal{D}_r^-)$;
// Parameter update
6 **for** $t = 1; t \leq iter; t++$ **do**
7 **foreach** $(u_p, v_i, v_k) \in \mathcal{D}_c$, *and* $(u_p, v_i, v_j) \in \mathcal{D}_r$ **do**
8 $\mathbf{u}_p^{(t)} \leftarrow \mathbf{u}_p^{(t-1)} - \gamma \nabla \mathcal{J}(\mathbf{u}_p)$;
9 $\mathbf{v}^{(t)} \leftarrow \mathbf{v}^{(t-1)} - \gamma \nabla \mathcal{J}(\mathbf{v}), \mathbf{v}$=$\{\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k, \mathbf{v}_g, \mathbf{v}_h\}$;
10 **for** $l = 1; l \leq |\mathcal{C}_v|; l++$ **do**
11 $\mathbf{c}_l^{(t)} \leftarrow \mathbf{c}_l^{(t-1)} - \gamma \nabla \mathcal{J}(\mathbf{c}_l)$;
12 **if** $\mathcal{J}$ *has converged* **then**
13 break;

ample, which can be approximated via negative sampling as follows:

$$P(v_k|(u_p, v_i), \Theta) = \sigma(u_p^T v_k' + v_i^T v_k')$$
$$\prod_{g=1}^{N} \mathbb{E}_{(u_p, v_i, v_g) \sim P(\mathcal{D}_c^-)} \sigma(-(u_p^T v_g' + v_i^T v_g'))$$

where $\sigma(x) = 1/(1 + exp(-x))$ is the sigmoid function; $\mathcal{D}_c^- = \mathcal{D}_r$ is the opposite triple set of $\mathcal{D}_c$; $P(\mathcal{D}_c^-)$ is a function randomly sampling instances from $\mathcal{D}_c^-$. N is the number of negative instances to be drawn per positive instance. The idea behind negative sampling is that we want to maximize the similarity between v_k and (u_p, v_i) and minimize the similarity between a randomly sampled item v_g and (u_p, v_i). In this way, we can approximately maximize $P(v_k|(u_p, v_i), \Theta)$.

Similarly, $P(v_i|(u_p, v_k), \Theta), P((\neg v_j, v_i)|u_p, \Theta)$ are also approximated via negative sampling. One issue we should deal with is that computing the numerators of the softmax function $P((\neg v_j, v_i)|u_p, \Theta)$ is also very expensive, as we have at least $\mathcal{O}(mn * \min(|\mathcal{I}_{u_1}^+|, \cdots, |\mathcal{I}_{u_m}^+|))$ training triples in $\mathcal{D}_r$, where $|\mathcal{I}_{u_m}^+|$ is the size of $\mathcal{I}_{u_m}^+$. We thus randomly sample user-specific ranking triples instead of using all the triples. The optimization process is shown in Algorithm 4, which is mainly composed of two steps, i.e., negative sampling (line 4-6), and parameter update (line 7-14).

Complexity Analysis. The computational time is mainly taken by evaluating the objective function $\mathcal{J}$ (i.e., the softmax functions) and updating the related variables. The time to compute $\mathcal{J}$ is $\mathcal{O}\big(d|\mathcal{D}_c| + d|\mathcal{D}_r|\big)$, where

d is the dimension of embeddings, and $|\mathcal{D}_c|, |\mathcal{D}_r|$ are the sizes of item co-rated triples and user-specific ranking triples, respectively. For all gradients $\nabla\mathcal{J}(\mathbf{u}_p), \nabla\mathcal{J}(\mathbf{v}_i), \nabla\mathcal{J}(\mathbf{c}_l)$, the computational time are $\mathcal{O}\big(d|\mathcal{D}_c| + d|\mathcal{D}_r|\big)$, $\mathcal{O}\big(d|\mathcal{D}_c| + d|\mathcal{D}_r|\big)$ and $\mathcal{O}\big(d(|\mathcal{D}_c| + |\mathcal{D}_r|)|\mathcal{C}_{v_i}|\big)$, respectively. Generally, $|\mathcal{C}_{v_i}|$ is no larger than 10 in real-world applications [233]. Hence, the overall computational complexity is $(\#iteration * \mathcal{O}(d|\mathcal{D}_c| + d|\mathcal{D}_r|))$. Specifically, $|\mathcal{D}_c| \leq mq(q-1)/2$, where $q = \max(|\mathcal{I}^+_{u_1}|, \cdots, |\mathcal{I}^+_{u_m}|)$. In real-world applications, q is typically small (e.g., power-law distribution). For $\mathcal{D}_r$, as illustrated before, we adopt the random sampling method to reduce its number. To sum up, MRLR is scalable to large datasets.

11.4 Experiments and Results

11.4.1 Experimental Setup

Datasets. We adopt the Amazon Web store data [151], which contains a series of datasets from various domains (e.g., clothing, electronics). To evaluate the effectiveness of MRLR, we choose four datasets, including Clothing, Electronics, Sports, Home. Besides user-item interactions, the datasets also include the categories that each item belongs to. We uniformly sample the datasets, to balance their sizes in the same order of magnitude for cross-dataset comparison. Table 11.1 reports the statistics of the datasets.

Datasets	#Users	#Items	#Ratings	#Categories
Clothing	29,550	50,677	181,993	1,764
Electronics	59,457	64,348	518,291	1,292
Sports	28,708	46,315	237,578	1,293
Home	37,884	50,948	313,871	2,002

Table 11.1: Statistics of the datasets.

Comparison Methods. We compare with seven state-of-the-art algorithms, 1) PMF [159]: probabilistic matrix factorization; 2) BPR [180]: Bayesian personalized ranking; 3) FM [179]: factorization machine incorporating item category. Since FM generally outperforms all the other LFM based algorithms, we only compare with FM; 4) Item2Vec [16]: item embedding based method; 5) Meta-Prod2Vec [215]: incorporates item category based on Item2Vec; 6) CoFactor [133]: jointly factorizes rating and item co-rated matrices; 7) User2Vec [83]: considers the user as a global context while learning item embedding; Besides, four variants of our framework are compared, a) RL:

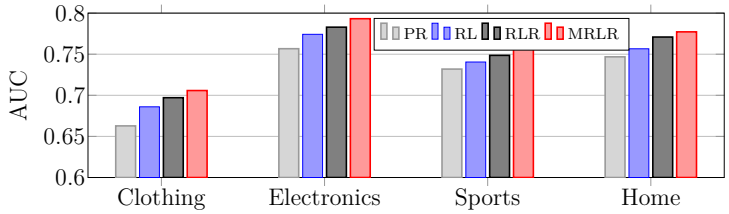


Figure 11.1: The results of our four variants.

the RL model considering only user and item embedding ; b) PR: personalized ranking model; c) RLR: the RL model combining a) and b); d) MRLR: multi-level RL model considering multi-level item organizations based on c).

Evaluation. Standard 5-fold cross validation is adopted to evaluate all the methods. The Area Under the ROC Curve (AUC) is used as the evaluation metric. Larger AUC indicates better recommendation performance.

Parameter Settings. The optimal parameter settings for all methods have been empirically estimated. We apply a grid search in $\{0.001, 0.01, 0.1, 1.0\}$ for the learning rate γ, regularization coefficient λ_Θ and 1/2-way regularization of FM, and a grid search in $\{1, 5, 10, 20, 50\}$ for the number of negative instances N. We set the dimension $d = 10$.

11.4.2 Results of MRLR

Results of Variants. The performance of our four variants is depicted in Figure 11.1. RLR outperforms both PR and RL – by 3.54% and 1.42% in AUC respectively (both significant, Paired t-test with p-value $< .01$), showing the effectiveness of both representation learning and personalized ranking. MRLR, which combines RLR with multi-level item organizations, performs the best among the four variants – with 1.12% lift in AUC compared to RLR (p-value $< .01$), indicating the benefit of considering fine-grained item relationships.

Impacts of Parameter α. Parameters α_1, α_2 control the importance of personalization and item co-occurrence relationships as shown in Equation 11.3. α_3 controls the effect of item category for adapting item embedding as shown in Equation 11.5. We apply a grid search ranging from 0 to 1 with step 0.1 to investigate their impacts. As $\alpha_1 + \alpha_2 = 1$, we only investigate the impacts of α_1, α_3 in our study. The results are described by Figure 11.2. For the four datasets, as α_1 varies from small to large, the performance first increases then decreases, with the maximum reached at around 0.8. This

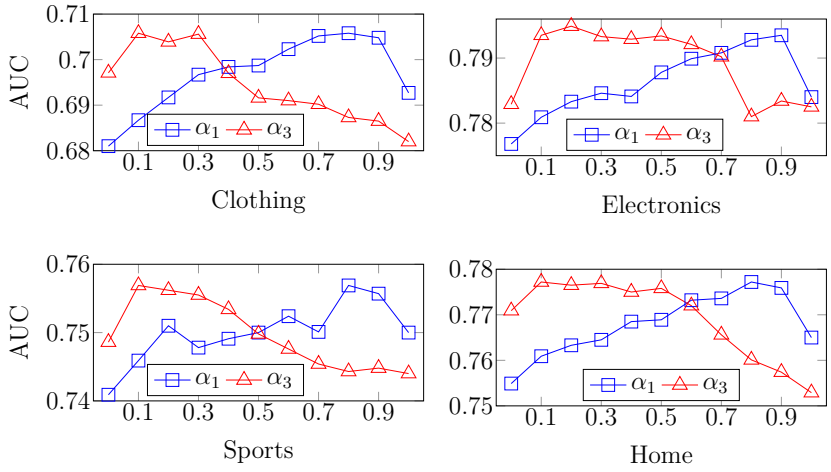


Figure 11.2: The effects of parameters α_1, α_3.

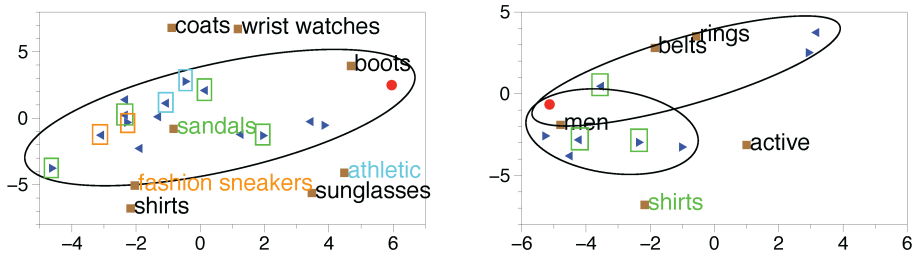


Figure 11.3: Visualization of user (red dot), item (blue triangle), and category (brown square) embeddings in a two dimensional space. Left-pointing triangles are rated items; right-pointing triangles are recommended items. The category of an item is labelled by a rectangle whose color is the same as its belonging category.

indicates that user preferences play an important role in item recommendation. In terms of α_3, we observe that the optimal settings range from 0.1 to 0.2, denoting a substantial contribution of item categories in recommendation. The similarity in performance variation across α_1, α_3 values on the four datasets demonstrates the robustness of MRLR.

Visualization of Embeddings. Our MRLR framework can generate meaningful embeddings that help interpret recommendation results. To show this, we visualize the embeddings of users, items and categories learnt by MRLR in a two dimensional space using t-SNE [144]. Figure 11.3 demonstrates the results of two examples in the Clothing dataset. For conciseness, here we do not visualize the other datasets, however, similar observations as below can be obtained: 1) the rated items and the recommended items are generally

clustered. This indicates certain similarity among the rated items by a user and the recommended items to the same user. 2) each cluster is located at the side of the user, and the user is represented as an endpoint of these clusters, indicating that user preference can be manifested as the direction along which the rated items are clustered. This suggests that the recommendations are determined by both rated items and user preferences. Finally, we note that the categories of recommended items are overlapped with those of the rated items. For instance, for user in the right plot the overlapped category is Shirts, indicating user preference over shirts. For user in the left plot the overlapped categories are Athletic, Fashion Sneakers, and Sandals, indicating that the user has a more diverse set of preferences. These observations show that MRLR can capture meaningful item relationships in multiple levels of item organizations – individual items, items in the same category, and items rated by the same user.

11.4.3 Comparative Results

Table 11.2 summarizes the performance of all comparison methods. Two views are considered: 'All Users' indicates all users are considered in the test data; while 'Cold Start' indicates only users with less than 5 ratings are involved in the test data. Several interesting findings are observed as follows.

Among the latent factor model based methods (PMF, BPR and FM), PMF performs the worst, as it is the basic rating prediction method without considering any auxiliary information. FM significantly outperforms PMF, indicating the effectiveness of item category for better recommendation. Interestingly, the performance of FM is worse than that of BPR. This verifies that personalized ranking is more effective than rating prediction in real-world recommendation scenarios.

The RL methods, including Item2Vec, MetaProd2Vec, CoFactor and User2Vec, generally perform better than latent factor based methods, despite being rating prediction models. This confirms that representation learning is more effective than latent factor models for recommendation. Among them, Item2Vec performs worse than MetaProd2Vec. This observation further confirms the previous conclusion that item category is useful to improve recommendation performance.

CoFactor and User2Vec consider personalization in addition to item embedding. CoFactor is equivalent to the CMF method as it simultaneously factorizes user-item and item-item co-occurrence matrices with shared item latent factors, while User2Vec adopts CBOW to integrate personalization.

Datasets	Cases	PMF	BPR	FM	Item2Vec	MP2Vec	CoFactor	User2Vec	MRLR	Improve
Clothing	All Users	0.5255	0.6151	0.5972	0.6429	0.6600*	0.6012	0.6249	**0.7058**	6.94%
	Cold Start	0.5291	0.6135	0.5969	0.6426	0.6602*	0.5984	0.6203	**0.7022**	6.36%
Electronics	All Users	0.6595	0.7178	0.7066	0.7529	0.7604*	0.7000	0.7121	**0.7932**	4.31%
	Cold Start	0.6558	0.7161	0.7010	0.7535	0.7631*	0.6937	0.7107	**0.7935**	3.98%
Sports	All Users	0.6136	0.6992	0.6856	0.7015	0.7148*	0.6693	0.6852	**0.7569**	5.89%
	Cold Start	0.6175	0.7013	0.6861	0.7063	0.7149*	0.6679	0.6883	**0.7541**	5.48%
Home	All Users	0.6319	0.6930	0.6795	0.7297	0.7455*	0.6737	0.6969	**0.7772**	4.25%
	Cold Start	0.6408	0.6911	0.6841	0.7317	0.7449*	0.6731	0.6917	**0.7763**	4.22%

Table 11.2: Performance (AUC) of comparison methods, where the best performance is highlighted in bold; the second best performance of other methods is marked by '*'; 'Improve' indicates the improvements of MRLR relative to the '*' results.

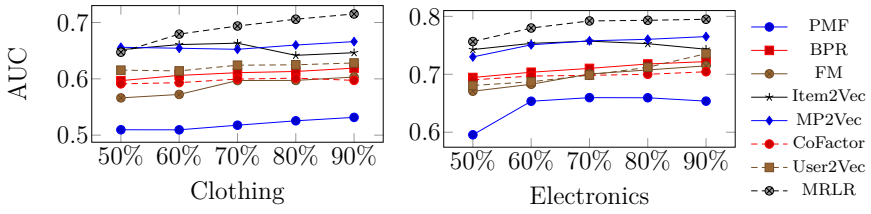


Figure 11.4: Impacts of data sparsity on the performance.

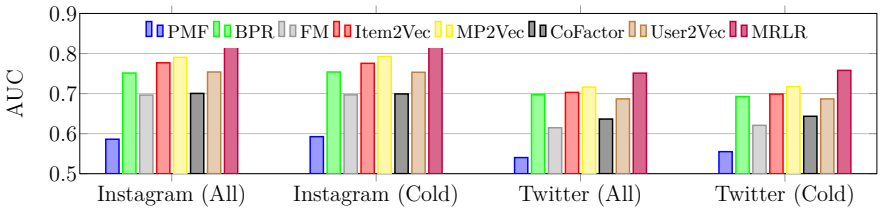


Figure 11.5: Comparative results on Instagram and Twitter.

Theoretically, the performance of the two methods should be better than that of Item2Vec, since they can provide users with personalized item list. We empirically find that User2Vec outperforms CoFactor, but both are slightly worse than Item2Vec. However, our proposed variant RL with Skip-gram outperforms Item2Vec, by 6.37% on average (Figure 11.1). Hence, we conjecture that considering personalization with Item2Vec helps improve recommendation performance, but CMF, CBOW are less effective than Skip-gram in incorporating item co-occurrence relationships with personalization.

Overall, compared with all the other methods, MRLR performs the best by learning user and item embeddings from a multi-level item organization, i.e., items in user-specific ranked list, items in the same category, and individual items. The improvements w.r.t. 'All User' and 'Cold Start' are 5.35%, 5.01% on average (both with p-value $< .01$), respectively. This implies that recommendation performance can be further enhanced by appropriately considering multi-level representation learning and personalized ranking.

Impacts of Data Sparsity. We further investigate the impacts of data sparsity on the performance of the comparison methods. Figure 11.4 depicts the variation of performance of all methods on Clothing & Electronics when the percentage of training data size w.r.t. the overall data size increases from 50% to 90%. We observe that MRLR consistently outperforms other methods across all levels of data sparsity. Furthermore, the performance of MRLR with data sparsity at 60% is better than that of any of other methods with data sparsity at 90%. Such observations also hold in other datasets,

which demonstrate that MRLR can achieve better performance even with high data sparsity.

Generalizability. To evaluate the generalizability of MRLR, we further collect data of Foursqure check-in performed over 3 weeks in 4 European capital cities (Amsterdam, London, Paris, Rome), published on Instagram (31,872 users perform 198,801 check-in at 41,387 locations that belong to 492 categories) and Twitter (18,522 users; 109,790 check-in; 38,855 locations; 482 categories). Figure 11.5 compares the performance of MRLR and the other methods. As in the previous setting, MRLR significantly outperforms (p-value $< .01$) the second best method MetaProd2Vec by 5.10% on 'All Users' and 5.62% on 'Cold Start'. These results show that MRLR can be effective in multiple recommendation tasks.

11.5 Conclusion

Representation learning (RL) has drawn much attention in recommendation, due to its effectiveness in capturing local item relationships. However, all existing RL based methods model recommendation as a rating prediction problem while recommendation is essentially a personalized ranking one. Besides, they all neglect multi-level organizations of items for fine-grained item relationships. Hence, this chapter proposes a multi-level RL framework for personalized ranking – MRLR, which learns user and item embeddings from a multi-level item organization for better recommendation. MRLR, therefore, benefits from RL as well as achieves the goal of personalized ranking. Empirical validation on real-world datasets shows that MRLR significantly outperforms state-of-the-art algorithms.

Chapter 12

Conclusion

Crowd knowledge creation has shown to be effective for knowledge generation at scale. It plays a central role in both on-line knowledge crowdsourcing systems (e.g., Wikipedia, StackOverflow) and human computation systems (e.g., Amazon MTurk, CrowdFlower). However, the theory and practice of crowd knowledge creation currently lacks a clear understanding on how to control the process for *efficiently* generating *high-quality* knowledge. This thesis has therefore been focused at developing novel methods and tools to improve knowledge creation processes, both in terms of the speed of generating knowledge and the quality of the generated knowledge.

We have tackled the problem from three angles: crowd modeling, task modeling, and task assignment. Each of the three angles is addressed by an individual part of this thesis. In this chapter, we summarize the main contributions of each part and provide an outlook to possible future research directions.

12.1 Summary of Contributions

12.1.1 Crowd Modeling

We model properties of crowds that are highly relevant for generating high-quality knowledge, namely, crowd expertise and preferences in knowledge creation. By studying expertise characterization along multiple dimensions, our work contributes new understanding to the current status, limitation, and potential of expertise usage in crowd knowledge creation. To fully exploit expertise, we further contribute insights on the relationships between crowd preferences and knowledge creation demand and outcomes.

Expertise Characterization. To better capture expertise, we contribute a new metric for expertise measurement, i.e. Mean Expertise Contribution (MEC), which captures expertise based on the quality of the contributed knowledge measured through social judgement and task difficulty. We conducted a comparative study to show how experts behave differently from other crowds in knowledge creation activities, and further showed that they are less influenced by gamification mechanisms.

To further our understanding on expertise, we improved our model by considering more refined traits of expertise, namely specialist expertise and ubiquitous expertise. We contribute a principled characterization of expertise along these two traits and their manifestations in both individual and social activities across multiple platforms. We showcased the applicability of the proposed expertise characterization in question routing.

Crowd Preferences and Market Dynamics. To understand crowds' preferences in knowledge creation, we categorized the discussions among crowd communities in fora. We showed that tasks of certain types (e.g., survey), and requesters with certain properties (e.g., those more generous) are more favoured by crowds. By analyzing the relationships between crowd discussions in fora and task executions in marketplaces, we provided evidence that crowd discussions are influenced by knowledge demand in the marketplace, and that crowd discussions on tasks can positive influence task completion rates in the marketplace.

12.1.2 Task Modeling

We modeled three properties of tasks that were found to be highly related to the quality and the speed of their executions, namely, the quality of task formulations, the complexity of tasks, and the clarity of tasks. For each

of them, we showed their effects on knowledge creation, and provided design methods for measuring their magnitude based on task features, and guidelines for better task design.

Quality of Task Formulations. Through a qualitative analysis on question edits in question-answering systems, we identified seven types of task edits that can significantly influence the quality of task formulations. To assist askers in question formulation, we proposed and executed two prediction tasks: 1) edit prediction, to detect whether a question needs an edit; and 2) edit type prediction, to automatically suggest edit types to improve the quality of task formulations.

Complexity of Tasks. To understand task complexity, we instrumented an experiment to collect workers' assessment on the complexity of real-world human computation tasks. We showed that task complexity is dependent on task types; and, while being subjective, it is coherently perceived across workers. We designed a high-dimensional regression model to measure task complexity based on three classes of structural features, including metadata features, content features, and visual features. We showed how different features affect the perception of task complexity, and how they can improve the performance of predicting task completion rates.

Clarity of Tasks. We surveyed workers in the CrowdFlower platform, which unveiled the concerns of crowds in confronting unclear tasks. By decomposing task clarity into two dimensions, i.e. goal clarity and role clarity, we showed that these two types of clarity are high-correlated with each other. Similar to our study on task complexity, we proposed a set of features to measure task clarity. We further showed that task clarity is uncorrelated with task complexity, thus contributing to a new understanding on the relationship between these two important task properties.

12.1.3 Task Assignment

To accurately associate tasks to crowds, we designed new task assignment methods that consider both properties of crowds and tasks. We introduced two methods, i.e. ReMF and HieVH, to fully exploit the properties of crowds and tasks by considering the relationships among the properties when they are organized in a hierarchy (e.g., taxonomy). We further advanced our work by developing a neural network based method, i.e. MRLR, to better learn the representations of crowds and tasks with their properties for task assignment.

Incorporating Crowd and Task Properties. We performed a long-term cross-topic analysis, showing that different crowd engagement properties, including intrinsic and extrinsic motivations, and expertise, are independent with each other. We integrated these crowd properties and task topics into task assignment, and demonstrated their positive effects on improving the performance of task assignment.

Exploiting the Structure of Crowd and Task Properties. We analyzed several real-world datasets where properties are organized in a hierarchy, and showed that vertically-affiliated properties of crowds (or tasks) in the hierarchy can be used to describe the similarity between crowds (or tasks). To model such similarity, we proposed a novel regularization method, i.e. recursive regularization. We further designed a novel recommendation method, i.e. ReMF, that integrates recursive regularization into the widely used matrix factorzation model for task assignment. Experimental validation showed that ReMF achieves a lift of approximately 10% in AUC compared with the state of the art.

We further investigated two other relationships among crowds or tasks, namely, complementary and alternative relationships. By defining metrics for measuring these relationships, we showed in real-world datasets that both relationships can be induced from the horizontal dimension of hierarchically-organized properties of crowds and tasks. We designed a novel method, i.e. HieVH, to account for both relationships to improve task assignment. Extensive validation showed that HieVH outperforms ReMF by more than 5% in AUC.

Learning Crowd and Task Representations. To exploit state-of-the-art neural network based methods for task assignment, we adapted the general representation learning method to predict the ranking of tasks. We designed a unified Bayesian framework, i.e. MRLR, to integrate task ranking with the structured properties of crowds and tasks for accurate task assignment, resulting in a lift of more than 5% in AUC compared with state-of-the-art methods.

12.2 Future Work

This thesis has contributed novel methods and findings for crowd knowledge creation acceleration from several important perspectives. However, there is still space for improvements. In this section, we will identify the limitations of the thesis and outline future work for further improve it. In addition,

we will look beyond crowd knowledge creation and investigate the broader area of knowledge creation. We will identify challenges and opportunities for which we believe could shape the research on knowledge creation in the future.

12.2.1 Improving Crowd Knowledge Creation

In the following we list future work for each of the three aspects of crowd knowledge creation acceleration addressed by this thesis.

Crowd Modeling. Part I shows that crowds often discuss intensively within communities in knowledge creation platforms or in related fora. To further our understanding on crowd properties for knowledge creation acceleration, it would be valuable to investigate the different roles individuals can play in communities and the possible relationships among individuals in communities [32]. Open research questions to address include: 1) whether or not some individuals lead the discussion, i.e. leaders, and the others follow the discussion, i.e. followers; 2) how to identify leaders and followers; and, 3) how leaders may influence followers in discussions and consequences in task executions. Going beyond the relationship among crowds, future work could also be focused on understanding the trust among different actors in knowledge creation, including crowd communities, requesters, and platform owners. Establishing trust among these actors is highly important for the success of knowledge creation and the sustainability of knowledge creation systems. Despite this, trust in state-of-the-systems is currently built based on fragment, opaque, and often incomplete knowledge. To improve the process of trust creation, crowd modeling is useful for creating open and extensible profiles as a basis for trust creation. More detailed discussions on the motivation, challenge, and future work can be found in our recent position paper [227].

Task Modeling. Following our study on the complexity and clarity of tasks in Part II, we would recommend future research to focus on a closely related yet different property, namely, task difficulty. All these three properties are important factors of cognitive load. While the difference between task clarity and the other two properties can be intuitively understood, i.e. clarity is more about the presentation and organization of task description (Sweller and Chandler [209]), the boundary between task complexity and task difficulty is less clear. Robinson [185] suggests to view task complexity as an invariant property of a task that only relates to task design, while viewing difficulty as a property relating to the joint effect of task structure and *human factors*, such as motivation, confidence, proficiency, intelligence, etc. Such a distinction

has been confirmed by our results: we found task complexity is coherently perceived across workers and can be accurately measured by task features. Following the theory of Robinson, future work could therefore be specifically focused on 1) the role of human factors and the relationship between them in task difficulty, and 2) the structural features of tasks in contributing to task difficulty and to the overall crowd cognitive load in task execution.

Task Assignment. Future work could be considered in two orthogonal directions to extend our work on task assignment addressed in Part III. On the one hand, efforts could be spent on incorporating richer properties of crowds and tasks for task assignment. Such work can be focused on either properties organized in more complex structures, e.g., graph structure (e.g., knowledge bases), or properties with unique representations, e.g., spatio-temporal properties. On the other hand, it would be valuable to consider more sophisticated methods to optimize the performance of task assignment. For example, deep neural networks have shown to be effective in modeling user behaviours based on interactional data [96], thus can be potentially useful to improve the accuracy of task assignment by modeling crowd behaviours in task executions. Furthermore, methods such as recurrent neural networks [190] can be applied to capturing the temporal dynamics of crowd behaviours, which may change over time due to the evolution of their expertise, preferences, etc.

12.2.2 The Future of Knowledge Creation

Knowledge creation can be achieved either by leveraging human intelligence or machine intelligence. Both approaches have been developing rapidly over the recent years. Given the complementary capabilities of humans and machines, we envision that knowledge creation can be further improved by combining the power of human and machine intelligence. In fact, while this thesis focuses on crowd knowledge creation, our work benefits a lot from the machine intelligence. We exploit machine intelligence to develop methods for accelerating crowd knowledge creation, e.g., we develop regression models for measuring task complexity and clarity, and latent factor models and neural network models for associating tasks to crowds. On the other hand, we could also take human intelligence as a means to assist machines to more effectively generate high-quality knowledge.

In order to be applied in a trusted manner, intelligent machines designed for autonomous knowledge generation require the following properties: 1) accountability, i.e., being able to explain how certain results are generated; and 2) transparency, i.e., allowing for inspection when the results are not

expected. By analyzing existing literature, we outline how humans could improve intelligent machines in these two aspects.

Accountable Machine Intelligence Models. The accountability of machine intelligence models has drawn an increasing amount of attention from research communities. Recently, Lipton [135] provides a requirement specification for model accountability from the following two perspectives: 1) model properties, which include simulatability, decomposability, and algorithmic transparency; and 2) techniques to allow *humans* to explain the exact process by which models work, such as verbal/textual explanation, visualization, and explanation by examples. Some efforts have been devoted to both directions. For example, Ribeiro et al. [182] propose a technique that approximates decisions made by complex models with low-complexity models (with simulatability), which facilitates human explanations with fidelity on how the actual model works. On the other hand, Change et al. [41] design a workflow for explaining recommendation models by tapping large-scale crowds. Despite these, it remains an open question how to best involve humans to solve the accountability problem for various machine intelligence models.

Debuggable Machine Intelligence Tasks. Programs implementing machine intelligence tasks are often used as black-boxes. Unlike traditional computer programs, these programs usually rely on large amount of data for model training. Moreover, the training process generally involves stochastic, approximate, search and optimization algorithms. Such black-box nature has made machine intelligence tasks particularly difficult to debug. To allow machine intelligence tasks to be debuggable, methods are expected to support the detection of two types of errors, namely, errors in model and training algorithms and errors in training data. In these methods, both automatic tools and humans are required: tools to support automatically locate possible sources of the error, and humans to verify the error. Recent work has started to address the problem from the perspective of tools. For example, Chakarov et al. [38] propose a technique to automatically identify training data points that cause misclassification in the test data. However, little work has explore the potential of crowds in debugging machine intelligence tasks. We envision that crowds have the potential to play an important role, especially in verifying the second type of errors (i.e., errors in training data), which can be done with little or no expertise on machine intelligence. In fact, the effectiveness of humans in justifying decisions and verifying issues has been proven in related domains [152, 20]. However, it remains an open question how to exploit crowds for debugging machine intelligence tasks.

Bibliography

[1] Andrew Abbott. The sociology of work and occupations. *Annual Review of Sociology*, pages 187–209, 1993.

[2] Fabian Abel, Nicola Henze, Eelco Herder, and Daniel Krause. Interweaving public user profiles on the web. In *Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization*, pages 16–27. Springer Berlin Heidelberg, 2010.

[3] Fabian Abel, Eelco Herder, Geert Jan Houben, Nicola Henze, and Daniel Krause. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2-3): 169–209, 2012.

[4] Deepak Agarwal and Bee-Chung Chen. Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

[5] June Ahn, Brian S Butler, Cindy Weng, and Sarah Webster. Learning to be a better q'er in social q&a sites: social norms and information artifacts. *Proceedings of the Association for Information Science and Technology*, 50(1):1–10, 2013.

[6] Amir Albadvi and Mohammad Shahbazi. A hybrid recommendation technique based on product category attributes. *Expert Systems with Applications*, 36(9):11480–11488, 2009.

[7] Omar Alonso and Ricardo Baeza-Yates. Design and implementation of relevance assessments using crowdsourcing. In *European Conference on Information Retrieval*, pages 153–164. Springer, 2011.

[8] Omar Alonso, Catherine Marshall, and Marc Najork. Crowdsourcing a subjective labeling task: a human-centered framework to ensure reliable results. Technical report, MSR-TR-2014-91, 2014.

[9] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 850–858, 2012.

[10] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. Steering user behavior with badges. In *Proceedings of the 22nd international conference on World Wide Web*, pages 95–106. International World Wide Web Conferences Steering Committee, 2013.

[11] Andrew Arnold, Yan Liu, and Naoki Abe. Temporal causal modeling with graphical granger methods. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 66–75. ACM, 2007.

[12] Muhammad Asaduzzaman, Ahmed Shah Mashiyat, Chanchal K. Roy, and Kevin A. Schneider. Answering Questions About Unanswered Questions of Stack Overflow. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, pages 97–100, 2013.

[13] S. Attfield, G. Kazai, M. Lalmas, and B. Piwowarski. Towards a science of user engagement (position paper). In *WSDM Workshop on User Modelling for Web Applications*, February 2011.

[14] Ali Sajedi Badashian, Afsaneh Esteki, Ameneh Gholipour, Abram Hindle, and Eleni Stroulia. Involvement, contribution and influence in github and stack overflow. In *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering*, pages 19–33, 2014.

[15] Krisztian Balog, Leif Azzopardi, and Maarten De Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–50. ACM, 2006.

[16] Oren Barkan and Noam Koenigstein. Item2vec: Neural item embedding for collaborative filtering. *IEEE Workshop on MLSP*, 2016.

[17] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003.

[18] Janine Berg. Income security in the on-demand economy: findings and policy lessons from a survey of crowdworkers. *Comparative Labor Law & Policy Journal*, 37(3), 2016.

[19] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. Cross-domain mediation in collaborative filtering. In *International Conference on User Modeling*, pages 355–359. Springer, 2007.

[20] Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. *Communications of the ACM*, 58(8):85–94, 2015.

[21] Jiang Bian, Yandong Liu, Ding Zhou, Eugene Agichtein, and Hongyuan Zha. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In *Proceedings of the 18th International Conference on World Wide Web*, pages 51–60. ACM, 2009.

[22] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[23] Stefano Bocconi, Alessandro Bozzon, Achilleas Psyllidis, Christiaan Titos Bolivar, and Geert-Jan Houben. Social glass: A platform for urban analytics and decision-making through heterogeneous social data. In *WWW Companion*, 2015.

[24] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1 – 8, 2011.

[25] Amiangshu Bosu, Christopher S Corley, Dustin Heaton, Debarshi Chatterji, Jeffrey C Carver, and Nicholas A Kraft. Building reputation in stackoverflow: an empirical investigation. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, pages 89–92. IEEE Press, 2013.

[26] Mohamed Bouguessa, Benoît Dumoulin, and Shengrui Wang. Identifying authoritative actors in question-answering forums: the case of yahoo! answers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 866–874. ACM, 2008.

[27] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[28] Alessandro Bozzon, Marco Brambilla, Stefano Ceri, and Silvia Quarteroni. A framework for integrating, exploring, and searching location-based web data. *IEEE Internet Computing*, 15(6):24–31, 2011.

[29] Alessandro Bozzon, Marco Brambilla, and Stefano Ceri. Answering search queries with crowdsearcher. In *Proceedings of the 21st International Conference on World Wide Web*, pages 1009–1018. ACM, 2012.

[30] Alessandro Bozzon, Marco Brambilla, Stefano Ceri, Matteo Silvestri, and Giuliano Vesci. Choosing the right crowd: expert finding in social networks. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 637–648. ACM, 2013.

[31] Alessandro Bozzon, Marco Brambilla, Stefano Ceri, Andrea Mauri, and Riccardo Volonterio. Pattern-based specification of crowdsourcing applications. In *International Conference on Web Engineering*, pages 218–235. Springer, 2014.

[32] Alessandro Bozzon, Piero Fraternali, Luca Galli, and Roula Karam. Modeling crowdsourcing scenarios in socially-enabled human computation applications. *Journal on Data Semantics*, 3(3):169–188, 2014.

[33] Hein Broekkamp, Bernadette HAM van Hout-Wolters, Gert Rijlaarsdam, and Huub van den Bergh. Importance in instructional text: teachers' and students' perceptions of task demands. *Journal of Educational Psychology*, 94(2):260, 2002.

[34] Francesco Buccafurri, Vincenzo Daniele Foti, Gianluca Lax, Antonino Nocera, and Domenico Ursino. Bridge analysis in a social internetworking scenario. *Information Science*, 224:1–18, 2013.

[35] Michael Burns and Xerxes Kotval. Questions About Questions: Investigating How Knowledge Workers Ask and Answer Questions. *Bell Labs Technical Journal*, 17(4):43–61, 2013.

[36] Donald J Campbell. Task complexity: A review and analysis. *Academy of Management Review*, 13(1):40–52, 1988.

[37] Francisco Cano and María Cardelle-Elawar. An integrated analysis of secondary school students' conceptions and beliefs about learning. *European Journal of Psychology of Education*, 19(2):167–187, 2004.

[38] Aleksandar Chakarov, Aditya Nori, Sriram Rajamani, Shayak Sen, and Deepak Vijaykeerthy. Debugging machine learning tasks. *arXiv preprint arXiv:1603.07292*, 2016.

[39] Jeanne Sternlicht Chall and Edgar Dale. *Readability revisited: The new Dale-Chall readability formula.* Brookline Books, 1995.

[40] Shuo Chang and Arnab Pal. Routing questions for collaborative answering in community question answering. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 494–501, 2013.

[41] Shuo Chang, F Maxwell Harper, and Loren Gilbert Terveen. Crowd-based personalized natural language explanations for recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 175–182. ACM, 2016.

[42] Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. Svdfeature: a toolkit for feature-based collaborative filtering. *Journal of Machine Learning Research*, 13(1):3619–3622, 2012.

[43] Justin Cheng, Jaime Teevan, and Michael S. Bernstein. Measuring crowdsourcing effort with error-time curves. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1365–1374. ACM, 2015a.

[44] Justin Cheng, Jaime Teevan, Shamsi T. Iqbal, and Michael S. Bernstein. Break it down: A comparison of macro- and microtasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 4061–4064. ACM, 2015b.

[45] Ta Chung Cheng, Chia Jui Lin, Chih Hung Lin, and Pu Jen Cheng. Person identification between different online social networks. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, pages 94–101, 2014.

[46] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z. Sui. Exploring millions of footprints in location sharing services. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2011.

[47] Michelene T.H. Chi, Robert Glaser, and Marshall J. Farr, editors. *The nature of expertise.* Psychology Press, 1 edition edition, 1998.

[48] Harry Collins and Robert Evans. *Rethinking expertise.* University Of Chicago Press, 2007.

[49] Kevyn Collins-Thompson. Computational assessment of text readability: A survey of current and future research. *ITL-International Journal of Applied Linguistics*, 165(2):97–135, 2014.

[50] Kevyn Collins-Thompson and James P Callan. A language modeling approach to predicting reading difficulty. In *HLT-NAACL*, pages 193–200, 2004.

[51] Constantinos K Coursaris, Sarah J Swierenga, and Ethan Watrall. An empirical investigation of color temperature and gender effects on web aesthetics. *Journal of Usability Studies*, 3(3):103–117, 2008.

[52] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 191–198. ACM, 2016.

[53] Scott A Crossley, Kristopher Kyle, and Danielle S McNamara. The tool for the automatic analysis of text cohesion (taaco): automatic assessment of local, global, and text cohesion. *Behavior research methods*, pages 1–11, 2015.

[54] Tove I Dahl, Margrethe Bals, and Anne Lene Turi. Are students' beliefs about knowledge and learning associated with their reported use of learning strategies? *British journal of educational psychology*, 75(2): 257–273, 2005.

[55] Edgar Dale and Jeanne S Chall. The concept of readability. *Elementary English*, 26(1):19–26, 1949.

[56] Orphée De Clercq, Véronique Hoste, Bart Desmet, Philip Van Oosten, Martine De Cock, and Lieve Macken. Using the crowd for readability prediction. *Natural Language Engineering*, 20(03), 2014.

[57] David Dearman and Khai N. Truong. Why Users of Yahoo!: Answers Do Not Answer Questions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 329–332, 2010.

[58] Edward Deci and Richard M Ryan. Self-determination theory. *Handbook of theories of social psychology*, page 416, 2008.

[59] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[60] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1): 143–177, 2004.

[61] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. Pick-a-crowd: tell me what you like, and i'll tell you what to do. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 367–374. International World Wide Web Conferences Steering Committee, 2013.

[62] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, and Philippe Cudré-Mauroux. Scaling-up the crowd: Micro-task pricing schemes for worker retention and latency improvement. In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[63] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G Ipeirotis, and Philippe Cudré-Mauroux. The dynamics of micro-task crowdsourcing: The case of amazon mturk. In *Proceedings of the 24th International Conference on World Wide Web*, pages 238–247. International World Wide Web Conferences Steering Committee, 2015.

[64] Michael Eichler. Causal inference in time series analysis. *Causality: Statistical Perspectives and Applications*, pages 327–354, 2012.

[65] Carsten Eickhoff and Arjen P de Vries. Increasing cheat robustness of crowdsourcing tasks. *Information Retrieval*, 16(2):121–137, 2013.

[66] K Anders Ericsson. *The Cambridge handbook of expertise and expert performance*. Cambridge University Press, 2006.

[67] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. *SIAM Journal on Discrete Mathematics*, 17(1):134–160, 2003.

[68] Karën Fort, Gilles Adda, and K Bretonnel Cohen. Amazon mechanical turk: Gold mine or coal mine? *Computational Linguistics*, 37(2):413–420, 2011.

[69] Thomas Fritz, Jingwen Ou, Gail C Murphy, and Emerson Murphy-Hill. A degree-of-knowledge model to capture source code familiarity. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*, pages 385–394. ACM, 2010.

[70] Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. A taxonomy of microtasks on the web. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, pages 218–223. ACM, 2014.

[71] Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. Understanding malicious behavior in crowdsourcing platforms: the case of online surveys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1631–1640. ACM, 2015.

[72] Ujwal Gadiraju, Jie Yang, and Alessandro Bozzon. Clarity is a worthwhile quality - on the role of task clarity in microtask crowdsourcing. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*. ACM, 2017. To appear.

[73] Snehal Neil Gaikwad, Durim Morina, Rohit Nistala, Megha Agarwal, Alison Cossette, Radhika Bhanu, Saiph Savage, Vishwajeet Narwal, Karan Rajpal, Jeff Regino, et al. Daemo: A self-governed crowdsourcing marketplace. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 101–102. ACM, 2015.

[74] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler, Steffen Rendle, and Lars Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *Proceedings of the 10th IEEE International Conference on Data Mining*, 2010.

[75] Huiji Gao, Jiliang Tang, and Huan Liu. gSCorr: modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM Conference on Information and Knowledge Management*, 2012.

[76] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. Content-aware point of interest recommendation on location-based social networks. In *The 29th AAAI Conference on Artificial Intelligence*, 2015.

[77] Yihan Gao and Aditya Parameswaran. Finish them!: Pricing algorithms for human computation. *Proceedings of the VLDB Endowment*, 7(14):1965–1976, 2014.

[78] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[79] Catherine Grady and Matthew Lease. Crowdsourcing document relevance assessment with mechanical turk. In *HLT-NAACL workshop on creating speech and language data with Amazon's mechanical turk*, pages 172–179. Association for Computational Linguistics, 2010.

[80] Arthur C Graesser, Danielle S McNamara, Max M Louwerse, and Zhiqiang Cai. Coh-metrix: analysis of text on cohesion and language. *Behavior research methods, instruments, & computers*, 36(2):193–202, 2004.

[81] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[82] Mary L Gray, Siddharth Suri, Syed S Ali, and Deepti Kulkarni. The crowd is a collaborative network. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work and Social Computing*, pages 34–147. ACM, 2016.

[83] Mihajlo Grbovic, Vladan Radosavljevic, Nemanja Djuric, Narayan Bhamidipati, Jaikit Savla, Varun Bhagwan, and Doug Sharp. E-commerce in your inbox: Product recommendations at scale. In *Proceedings of the 21st ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1809–1818. ACM, 2015.

[84] Jinwen Guo, Shengliang Xu, Shenghua Bao, and Yong Yu. Tapping on the potential of q&a community by recommending answer providers. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 921–930. ACM, 2008.

[85] Ido Guy, Uri Avraham, David Carmel, Sigalit Ur, Michal Jacovi, and Inbal Ronen. Mining expertise and interests from social media. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 515–526, 2013.

[86] AF Hadwin, M Oshige, M Miller, and P Wild. Examining student and instructor task perceptions in a complex engineering design task. In *international conference on innovation and practices in engineering design and engineering education. McMaster University, Hamilton, ON, Canada*, 2009.

[87] Allison Hadwin. Student task understanding. In *Learning and Teaching Conference. University of Victoria, Victoria, British Columbia, Canada.*, 2006.

[88] Martin Halvey and Robert Villa. Evaluating the effort involved in relevance assessments for images. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 887–890. ACM, 2014.

[89] Benjamin V. Hanrahan, Gregorio Convertino, and Les Nelson. Modeling problem difficulty and expertise in stackoverflow. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work Companion*, pages 91–94. ACM, 2012.

[90] F. Maxwell Harper, Daphne Raban, Sheizaf Rafaeli, and Joseph A. Konstan. Predictors of Answer Quality in Online Q&A Sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 865–874, 2008.

[91] Simon Harper, Eleni Michailidou, and Robert Stevens. Toward a definition of visual complexity as an implicit measure of cognitive load. *ACM Transactions on Applied Perception*, 6(2):10, 2009.

[92] Sandra G Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 50, pages 904–908. Sage Publications, 2006.

[93] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in Psychology*, 52:139–183, 1988.

[94] David Hasler and Sabine E Suesstrunk. Measuring colorfulness in natural images. In *Electronic Imaging*, pages 87–95. International Society for Optics and Photonics, 2003.

[95] Ruining He, Chunbin Lin, Jianguo Wang, and Julian McAuley. Sherlock: sparse hierarchical embeddings for visually-aware one-class collaborative filtering. In *Proceedings of 25th International Joint Conference on Artificial Intelligence*, 2016.

[96] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.

[97] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.

[98] T Hoßfeld, Raimund Schatz, and Sebastian Egger. Sos: The mos is not enough! In *Proceedings of the Third International Workshop on Quality of Multimedia Experience*, pages 131–136. IEEE, 2011.

[99] Tobias Hoßfeld, Christian Keimel, Matthias Hirth, Bruno Gardlo, Julian Habigt, Klaus Diepold, and Phuoc Tran-Gia. Best practices for qoe crowdtesting: Qoe assessment with crowdsourcing. *IEEE Transactions on Multimedia*, 16(2):541–558, 2014.

[100] Jeff Howe. The rise of crowdsourcing. *Wired Magazine*, 14(6):1–4, 2006.

[101] Feng-Hsiung Hsu. *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*. Princeton University Press, 2002.

[102] Zhiting Hu, Poyao Huang, Yuntian Deng, Yingkai Gao, and Eric P Xing. Entity hierarchy embedding. In *ACL-IJCNLP*, 2015.

[103] Takashi Iba, Keiichi Nemoto, Bernd Peters, and Peter A. Gloor. Analyzing the Creative Editing Behavior of Wikipedia Editors: Through Dynamic Social Network Analysis. *Procedia - Social and Behavioral Sciences*, 2(4):6441 – 6456, 2010.

[104] Panagiotis G Ipeirotis. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, 17(2): 16–21, 2010.

[105] Lilly C Irani and M Silberman. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 611–620. ACM, 2013.

[106] Melody Y Ivory, Rashmi R Sinha, and Marti A Hearst. Empirically validated web page design metrics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 53–60. ACM, 2001.

[107] Diane Lee Jamieson-Noel. *Exploring task definition as a facet of self-regulated learning*. PhD thesis, Faculty of Education-Simon Fraser University, 2004.

[108] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20 (4):422–446, 2002.

[109] Rodolphe Jenatton, Julien Mairal, Francis R Bach, and Guillaume R Obozinski. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[110] Hyun Joon Jung and Matthew Lease. A discriminative approach to predicting assessor accuracy. In *European Conference on Information Retrieval*, pages 159–171. Springer, 2015.

[111] Hyun Joon Jung and Matthew Lease. Modeling temporal crowd work quality with limited supervision. *Proceedings of the 3rd AAAI Conference on Human Computation*, 2015.

[112] Pawel Jurczyk and Eugene Agichtein. Discovering authorities in question answer communities by using link analysis. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 919–922. ACM, 2007.

[113] Huzefa Kagdi, Maen Hammad, and Jonathan I Maletic. Who can help me with this source code change? In *IEEE International Conference on Software Maintenance*, pages 157–166. IEEE, 2008.

[114] Eirini Kalliamvakou, Georgios Gousios, Kelly Blincoe, Leif Singer, Daniel M. German, and Daniela Damian. In *Proceedings of the 11th Working Conference on Mining Software Repositories*.

[115] Bhargav Kanagal, Amr Ahmed, Sandeep Pandey, Vanja Josifovski, Jeff Yuan, and Lluis Garcia-Pueyo. Supercharging recommender systems using taxonomies for learning user purchase behavior. *Proceedings of the Very Large Data Base Endowment*, 5(10):956–967, 2012.

[116] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 79–86. ACM, 2010.

[117] Rohit J Kate, Xiaoqiang Luo, Siddharth Patwardhan, Martin Franz, Radu Florian, Raymond J Mooney, Salim Roukos, and Chris Welty. Learning to predict readability using diverse linguistic features. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 546–554. Association for Computational Linguistics, 2010.

[118] Shashank Khanna, Aishwarya Ratan, James Davis, and William Thies. Evaluating and improving the usability of mechanical turk for low-income workers in india. In *Proceedings of the first ACM Symposium on Computing for Development*, page 12. ACM, 2010.

[119] Seyoung Kim and Eric P Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *In Proceedings of the 27th International Conference on Machine Learning*, 2010.

[120] J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, DTIC Document, 1975.

[121] Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 453–456. ACM, 2008.

[122] Aniket Kittur, Jeffrey V Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. The future of crowd work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, pages 1301–1318. ACM, 2013.

[123] Noam Koenigstein, Gideon Dror, and Yehuda Koren. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the 5th ACM Conference on Recommender Systems*, 2011.

[124] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM Conference on Knowledge Discovery and Data Mining*, pages 426–434. ACM, 2008.

[125] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[126] Rochelle Laplante and M. Six Silberman. Building trust in crowd worker forums: Worker ownership, governance, and work outcomes. In *Weaving Relations of Trust in Crowd Work: Transparency and Reputation across Platforms. Workshop at ACM WebSci*, 2016.

[127] Edith Law and Luis von Ahn. Human computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(3):1–121, 2011.

[128] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196, 2014.

[129] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[130] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, 2001.

[131] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural information Processing Systems*, pages 2177–2185, 2014.

[132] Wu-Jun Li and Dit-Yan Yeung. Relation regularized matrix factorization. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009.

[133] Dawen Liang, Jaan Altosaar, Laurent Charlin, and David M Blei. Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 59–66. ACM, 2016.

[134] Christoph Lippert, Stefan Hagen Weber, Yi Huang, Volker Tresp, Matthias Schubert, and Hans-Peter Kriegel. Relation prediction in multi-relational domains using matrix factorization. In *NIPS Workshop SISO*, 2008.

[135] Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

[136] Lars Lischke, Sven Mayer, Katrin Wolf, Niels Henze, Albrecht Schmidt, Svenja Leifert, and Harald Reiterer. Using space: Effect of display size on users' search performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1845–1850. ACM, Apr 2015. ISBN 978-1-4503-3146-3.

[137] Jing Liu, Young-In Song, and Chin-Yew Lin. Competition-based user expertise score estimation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 425–434. ACM, 2011.

[138] Jing Liu, Quan Wang, Chin-Yew Lin, and Hsiao-Wuen Hon. Question Difficulty Estimation in Community Question Answering Services. In *Empirical Methods on Natural Language Processing*, pages 85–90, 2013.

[139] Xiaoyong Liu, W Bruce Croft, and Matthew Koll. Finding experts in community-based question-answering services. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 315–316, 2005.

[140] Kai Lu, Guanyuan Zhang, Rui Li, Shuai Zhang, and Bin Wang. Exploiting and exploring hierarchical structure in music recommendation. In *Information Retrieval Technology*, pages 211–225. Springer, 2012.

[141] Lieve Luyten, Joost Lowyck, and Francis Tuerlinckx. Task perception as a mediating variable: A contribution to the validation of instructional knowledge. *British Journal of Educational Psychology*, 71(2): 203–223, 2001.

[142] David Ma, David Schuler, Thomas Zimmermann, and Jonathan Sillito. Expert recommendation with usage expertise. In *IEEE International Conference on Software Maintenance*, pages 535–538, 2009.

[143] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, 2011.

[144] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[145] David Malvern and Brian Richards. Measures of lexical richness. *The Encyclopedia of Applied Linguistics*, 2012.

[146] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. Design Lessons from the Fastest Q&a Site in the West. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2857–2866, 2011.

[147] Catherine C Marshall and Frank M Shipman. Experiences surveying the crowd: Reflections on methods, participation, and reliability. In *Proceedings of the 5th Annual ACM Web Science Conference*, pages 234–243. ACM, 2013.

[148] David Martin, Benjamin V Hanrahan, Jacki O'Neill, and Neha Gupta. Being a turker. In *Proceedings of the 17th ACM Conference on Computer-Supported Cooperative Work and Social Computing*, pages 224–235. ACM, 2014.

[149] Andreu Mas-Colell, Michael Dennis Whinston, Jerry R Green, et al. *Microeconomic theory*, volume 1. Oxford University Press New York, 1995.

[150] Julian McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2015.

[151] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.

[152] Tyler McDonnell, Matthew Lease, Tamer Elsayad, and Mucahid Kutlu. Why is that relevant? collecting annotator rationales for relevance judgments. In *Proceedings of the 4th AAAI Conference on Human Computation and Crowdsourcing*, 2016.

[153] Brian McInnis, Dan Cosley, Chaebong Nam, and Gilly Leshed. Taking a hit: Designing around rejection, mistrust, risk, and workers' experiences in amazon mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2271–2282. ACM, 2016.

[154] Aditya Krishna Menon, Krishna-Prasad Chitrapura, Sachin Garg, Deepak Agarwal, and Nagaraj Kota. Response prediction using collaborative filtering with hierarchies and side-information. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 141–149. ACM, 2011.

[155] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[156] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing System*, pages 3111–3119, 2013.

[157] Tanushree Mitra, CJ Hutto, and Eric Gilbert. Comparing person- and process-centric strategies for obtaining quality data on amazon mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1345–1354. ACM, 2015.

[158] Andriy Mnih. Taxonomy-informed latent factor models for implicit feedback. In *KDD Cup*, pages 169–181, 2012.

[159] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Neural Information Processing Systems*, pages 1257–1264, 2007.

[160] Yashar Moshfeghi, Benjamin Piwowarski, and Joemon M Jose. Handling data sparsity in collaborative filtering using emotion and semantic based features. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011.

[161] Kevin Kyung Nam, Mark S Ackerman, and Lada A Adamic. Questions in, knowledge in?: a study of naver's question answering community. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 779–788. ACM, 2009.

[162] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pages 173–187. IEEE, 2009.

[163] S.M. Nasehi, J. Sillito, F. Maurer, and C. Burns. What makes a good code example?: A study of programming Q&A in Stack Overflow. In *IEEE International Conference on Software Maintenance*, pages 25–34, 2012.

[164] Jasper Oosterman, Archana Nottamkandath, Chris Dijkshoorn, Alessandro Bozzon, Geert-Jan Houben, and Lora Aroyo. Crowdsourcing knowledge-intensive tasks in cultural heritage. In *Proceedings of the 2014 ACM conference on Web science*, pages 267–268. ACM, 2014.

[165] Aditya Pal and Joseph A Konstan. Expert identification in community question answering: exploring question selection bias. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM'10, pages 1505–1508. ACM, 2010.

[166] Aditya Pal, Rosta Farzan, Joseph A Konstan, and Robert E Kraut. Early detection of potential experts in question answering communities. In *User Modeling, Adaption and Personalization*, pages 231–242. Springer, 2011.

[167] Aditya Pal, Shuo Chang, and Joseph A. Konstan. Evolution of Experts in Question Answering Communities. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, pages 274–281, 2012.

[168] Aditya Pal, Shuo Chang, and Joseph A Konstan. Evolution of experts in question answering communities. In *Proceedings of the Sixth International Conference on Weblogs and Social Media*, 2012.

[169] Aditya Pal, F Maxwell Harper, and Joseph A Konstan. Exploring question selection bias to identify experts and potential experts in community question answering. *ACM Transactions on Information Systems*, 30(2):10, 2012.

[170] Aditya Pal, Fei Wang, Michelle X Zhou, Jeffrey Nichols, and Barton A Smith. Question routing to user communities. In *Proceedings of the 22nd ACM Conference on Information and Knowledge Management*, pages 2357–2362. ACM, 2013.

[171] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Empirical Methods on Natural Language Processing*, volume 14, pages 1532–43, 2014.

[172] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Causal inference on time series using restricted structural equation models. In *Advances in Neural Information Processing Systems*, pages 154–162, 2013.

[173] Emily Pitler and Ani Nenkova. Revisiting readability: A unified framework for predicting text quality. In *Empirical Methods on Natural Language Processing*, pages 186–195. Association for Computational Linguistics, 2008.

[174] Achilleas Psyllidis, Alessandro Bozzon, Stefano Bocconi, and Christiaan Titos Bolivar. A platform for urban analytics and semantic data integration in city planning. In *International Conference on Computer-Aided Architectural Design Futures*, pages 21–36. Springer, 2015.

[175] Lizhen Qu, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, and Timothy Baldwin. Named entity recognition for novel types by transfer learning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2016.

[176] Sujith Ravi, Bo Pang, Vibhor Rastogi, and Ravi Kumar. Great question! question quality in community q&a. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, pages 426—435. AAAI, 2014.

[177] Judith Redi and Isabel Povoa. Crowdsourcing for rating image aesthetic appeal: Better a paid or a volunteer crowd? In *CrowdMM*, pages 25–30. ACM, 2014.

[178] Katharina Reinecke, Tom Yeh, Luke Miratrix, Rahmatri Mardiko, Yuechen Zhao, Jenny Liu, and Krzysztof Z Gajos. Predicting users' first impressions of website aesthetics with a quantification of perceived visual complexity and colorfulness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2049–2058. ACM, 2013.

[179] Steffen Rendle. Factorization machines. In *Proceedings of the 10th IEEE International Conference on Data Mining*, 2010.

[180] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[181] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011.

[182] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[183] Francesco Ricci, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

[184] Presentacion Rivera-Reyes. Students' task interpretation and conceptual understanding in electronics laboratory work. 2015.

[185] Peter Robinson. Task complexity, task difficulty, and task production: Exploring interactions in a componential framework. *Applied linguistics*, 22(1):27–57, 2001.

[186] Jakob Rogstadius, Vassilis Kostakos, Aniket Kittur, Boris Smus, Jim Laredo, and Maja Vukovic. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. In *Proceedings of the 5th International Conference on Weblogs and Social Media*, 2011.

[187] Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson. Who are the crowdworkers?: Shifting demographics in mechanical turk. In *SIGCHI Extended Abstracts on Human Factors in Computing Systems*, pages 2863–2872, New York, NY, USA, 2010. ACM.

[188] M. Rosvall and C. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.*, 105(1118), 2008.

[189] Libby O Ruch and Rae R Newton. Sex characteristics, task clarity, and authority. *Sex Roles*, 3(5):479–494, 1977.

[190] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5 (3):1, 1988.

[191] Shaghayegh Sahebi and Peter Brusilovsky. It takes two to tango: an exploration of domain pairs for cross-domain collaborative filtering. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 131–138. ACM, 2015.

[192] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing System*, volume 20, pages 1–8, 2011.

[193] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer, 2011.

[194] Bracha Shapira, Lior Rokach, and Shirley Freilikhman. Facebook single and cross domain data for recommendation systems. *User Modeling and User-Adapted Interaction*, 23(2-3):211–247, 2013.

[195] Aaron D Shaw, John J Horton, and Daniel L Chen. Designing incentives for inexpert human raters. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work*, pages 275–284. ACM, 2011.

[196] Aashish Sheshadri and Matthew Lease. Square: A benchmark for research on computing crowd consensus. In *Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[197] Yue Shi, Martha Larson, and Alan Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 269–272. ACM, 2010.

[198] Yue Shi, Alexandros Karatzoglou, Linas Baltrunas, Martha Larson, Nuria Oliver, and Alan Hanjalic. Climf: learning to maximize reciprocal rank with collaborative less-is-more filtering. In *Proceedings of the 6th ACM Conference on Recommender Systems*, pages 139–146. ACM, 2012.

[199] Yue Shi, Martha Larson, and Alan Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys*, 47(1):3, 2014.

[200] M Silberman, Lilly Irani, and Joel Ross. Ethics and tactics of professional crowdwork. 17(2):39–43, 2010.

[201] Giuseppe Silvestri, Jie Yang, Alessandro Bozzon, and Andrea Tagarelli. Linking accounts across social networks: the case of stackoverflow, github and twitter. In *Proceedings of the 1st International Workshop on Knowledge Discovery on the Web*, Cagliari, Italy, September 2015.

[202] Ajit P Singh and Geoffrey J Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 650–658. ACM, 2008.

[203] Donna Spencer. *Card sorting: Designing usable categories*. Rosenfeld Media, 2009.

[204] Laurentiu Catalin Stanculescu, Alessandro Bozzon, Robert-Jan Sips, and Geert-Jan Houben. Work and play: An experiment in enterprise gamification. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 346–358. ACM, 2016.

[205] Harald Steck. Evaluation of recommendations: rating-prediction and ranking. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 213–220. ACM, 2013.

[206] Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4, 2009.

[207] Zhu Sun, Jie Yang, Jie Zhang, and Alessandro Bozzon. Exploiting both vertical and horizontal dimensions of feature hierarchy for effective recommendation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 189–195. AAAI, 2017.

[208] Zhu Sun*, Jie Yang*, Jie Zhang, Alessandro Bozzon, Chi Xu, and Yu Chen. MRLR: Multi-level representation learning for personalized ranking in recommendation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017. (*: Joint first authors.) To appear.

[209] John Sweller and Paul Chandler. Why some material is difficult to learn. *Cognition and instruction*, 12(3):185–233, 1994.

[210] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. Exploiting homophily effect for trust prediction. In *Proceedings of the 6th ACM Conference on Web Search and Data Mining*, pages 53–62. ACM, 2013.

[211] Jiliang Tang, Xia Hu, and Huan Liu. Social recommendation: a review. *Social Network Analysis and Mining*, 3(4):1113–1133, 2013.

[212] Yla R. Tausczik, Aniket Kittur, and Robert E. Kraut. Collaborative Problem Solving: A Study of MathOverflow. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 355–367, 2014.

[213] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[214] C. Treude, O. Barzilay, and M. Storey. How do programmers ask and answer questions on the web?: NIER track. In *Proceedings of the ACM/IEEE International Conference on Software Engineering*, pages 804–807, 2011.

[215] Flavian Vasile, Elena Smirnova, and Alexis Conneau. Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 225–232. ACM, 2016.

[216] Bogdan Vasilescu, Vladimir Filkov, and Alexander Serebrenik. Stackoverflow and github: associations between software development and crowdsourced knowledge. In *Proceedings of the 2013 ASE/IEEE International Conference on Social Computing. IEEE*, pages 188–195, 2013.

[217] Bogdan Vasilescu, Alexander Serebrenik, Prem Devanbu, and Vladimir Filkov. How Social Q&A Sites Are Changing Knowledge Sharing in Open Source Software Communities. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 342–354, 2014.

[218] Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 155–164. ACM, 2014.

[219] Maja Vukovic, Jim Laredo, and Sriram Rajagopal. Challenges and experiences in deploying enterprise crowdsourcing service. In *International Conference on Web Engineering*, pages 460–467. Springer, 2010.

[220] Benjamin N. Waber and Alex Pentland. Recognizing expertise. In *Winter Conference on Business Intelligence*, University of Utah, Utah, USA, 2009.

[221] Christoph Wagner, Vera Liao, Peter Pirolli, Lynn Nelson, and Markus Strohmaier. It's not in their tweets: modeling topical expertise of Twitter users. In *Proceedings of the International Conference on Privacy, Security, Risk and Trust, and International Conference on Social Computing*, pages 91–100. IEEE, 2012.

[222] Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu. Exploring implicit hierarchical structures for recommender systems. In *Proceedings of the 27st International Joint Conference on Artificial Intelligence*, 2015.

[223] Markus Weimer, Alexandros Karatzoglou, Quoc Viet Le, and Alex Smola. Maximum margin matrix factorization for collaborative ranking. In *Advances in Neural Information Processing System*, pages 1–8, 2007.

[224] Li-Tung Weng, Yue Xu, Yuefeng Li, and Richi Nayak. Exploiting item taxonomy for solving cold-start problem in recommendation making. In *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence*, 2008.

[225] Robert E Wood. Task complexity: Definition of the construct. *Organizational Behavior and Human Decision Processes*, 37(1):60 – 82, 1986.

[226] Baoguo Yang and Suresh Manandhar. Tag-based expert recommendation in community question answering. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 960–963, 2014.

[227] Jie Yang and Alessandro Bozzon. On the improvement of quality and reliability of trust cues in micro-task crowdsourcing. In *Weaving Relations of Trust in Crowd Work: Transparency and Reputation across Platforms. Workshop at ACM WebSci*, 2016.

[228] Jie Yang, Claudia Hauff, Alessandro Bozzon, and Geert-Jan Houben. Asking the right question in collaborative Q&A systems. In *Proceedings of the 25th ACM conference on Hypertext and Social Media*, pages 179–189. ACM, 2014.

[229] Jie Yang, Ke Tao, Alessandro Bozzon, and Geert-Jan Houben. Sparrows and owls: Characterisation of expert behaviour in stackoverflow. In *Proceedings of the 22nd International Conference on User Modeling, Adaptation, and Personalization*, pages 266–277. Springer, 2014.

[230] Jie Yang, Alessandro Bozzon, and Geert-Jan Houben. Harnessing engagement for knowledge creation acceleration in collaborative q&a systems. In *Proceedings of the 23rd International Conference on User Modeling, Adaptation, and Personalization*, pages 315–327. Springer, 2015.

[231] Jie Yang, Alessandro Bozzon, and Geert-Jan Houben. Knowledge crowdsourcing acceleration. In *Proceedings of the 15th International Conference on Web Engineering*, pages 639–643. Springer, 2015.

[232] Jie Yang, Judith Redi, Gianluca Demartini, and Alessandro Bozzon. Modeling task complexity in crowdsourcing. In *Proceedings of the 4th AAAI Conference on Human Computation and Crowdsourcing*, pages 249–258. AAAI, 2016.

[233] Jie Yang, Zhu Sun, Alessandro Bozzon, and Jie Zhang. Learning hierarchical feature influence for recommendation by recursive regularization. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 51–58. ACM, 2016.

[234] Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen. CQArank: Jointly Model Topics and Expertise in Community Question Answering. In *Proceedings of the*

22nd ACM International Conference on Conference on Information and Knowledge Management, pages 99–108, 2013.

[235] Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen. Cqarank: jointly model topics and expertise in community question answering. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 99–108. ACM, 2013.

[236] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR Conference on Research & Development in Information Retrieval*, 2011.

[237] Reyyan Yeniterzi and Jamie Callan. Analyzing bias in cqa-based expert finding test sets. In *Proceedings of the 37th international ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 967–970. ACM, 2014.

[238] Ming Yin, Mary L Gray, Siddharth Suri, and Jennifer Wortman Vaughan. The communication network within the crowd. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1293–1303. International World Wide Web Conferences Steering Committee, 2016.

[239] Jing Zhang, Jie Tang, and Juanzi Li. Expert finding in a social network. In *Advances in Databases: Concepts, Systems and Applications*, pages 1066–1069. Springer, 2007.

[240] Jun Zhang, Mark S Ackerman, and Lada Adamic. Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web*, pages 221–230. ACM, 2007.

[241] Yuchen Zhang, Amr Ahmed, Vanja Josifovski, and Alexander Smola. Taxonomy discovery for personalized recommendation. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pages 243–252. ACM, 2014.

[242] Tong Zhao, Julian McAuley, and Irwin King. Improving latent factor models via personalized feature projection for one class recommendation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 821–830. ACM, 2015.

[243] Yi Zhen, Wu-Jun Li, and Dit-Yan Yeung. Tagicofi: tag informed collaborative filtering. In *Proceedings of the 3rd ACM Conference on Recommender Systems*, pages 69–76. ACM, 2009.

[244] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. *arXiv preprint arXiv:1706.05075*, 2017.

[245] Yanhong Zhou, Gao Cong, Bin Cui, Christian S Jensen, and Junjie Yao. Routing questions to the right users in online communities. In *Proceedings of the 25th IEEE International Conference on Data Engineering*, pages 700–711. IEEE, 2009.

[246] Cai-Nicolas Ziegler, Georg Lausen, and Lars Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, pages 406–415. ACM, 2004.

List of Figures

List of Tables

Summary

Crowd Knowledge Creation Acceleration

Crowd knowledge creation plays a central role in many types of Web based information systems, ranging from community question-answering (CQA) systems (e.g. StackOverflow and Quora) to micro-task crowdsourcing systems (e.g. Amazon mTurk and CrowdFlower). In these systems, knowledge demands are generally fulfilled by means of tasks (e.g. questions in CQA systems, micro-tasks in crowdsourcing systems) executed by group of individuals (e.g. contributors in CQA systems, workers in crowdsourcing systems). Despite of the success in some platforms, knowledge creation tasks so far are assumed to be of low cognitive complexity and are generally solved as a bottom-up process; as a consequence, outcomes are heavily dependent on the spontaneous and autonomous contribution of crowds. This limits our ability to control the volume, speed, and quality of knowledge creation. By unlocking the value of features related to human knowledge, e.g. expertise and motivation, we envision that crowd knowledge creation can reach its full potential where complex, cognitively intensive tasks are solved and thus high-quality knowledge is efficiently generated.

This thesis therefore focuses on understanding crowd knowledge creation processes and developing methods and tools for controlling and accelerating the process. To capture for this objective the key steps in crowd knowledge creation, we frame the discussion around a generic model [231], which builds on the following key components: ***1)*** *Crowd modeling* techniques, to assess features related to the knowledge of crowds; ***2)*** *Task modeling* techniques, to represent knowledge demands and resources for knowledge creation; ***3)*** *Task assignment* methods, for associating tasks to crowds. Our work aims at showing how, by optimally designing each of the components described above, it will be possible to accelerate crowd knowledge creation in a principled and effective way.

The first part of the thesis introduces our work on crowd modeling. We contribute new understanding and metrics for expertise, which is an important, multifaceted property of individuals and communities. Inspired by theories of expertise in sociology, we proposed a novel expertise metric based on social judgment in Chapter 2 (based on [229]), namely the Mean Expertise Contribution (MEC). By comparing with existing expertise metrics, we empirically found that MEC can better characterize crowd expertise than traditional metrics that are biased towards activeness. To account for the multi-dimensional manifestation of expertise in knowledge creation, resource sharing, and social interactions, we then extended our study to cross-platform expertise characterization in Chapter 3. We showed different crowd activities across-platforms can help characterize different expertise traits, i.e., specialist expertise and ubiquitous expertise. Next, we analyzed activities of crowds as communities, and showed how community activities could influence and be influenced by the dynamics of knowledge creation marketplaces in Chapter 4. Our results shed a new light on the importance of social and work-related preferences of communities in affecting task performance.

The second part of the thesis focuses on task modeling. We studied a set of task properties that can be related to the quality and speed of their execution by crowds, namely: the quality of task formulation, the complexity of tasks, and the clarity of tasks. Motivated by the large portion of poorly formulated tasks, we first analyzed task content factors that could substantially influence task quality in Chapter 5 (based on [228]). A qualitative study revealed seven important content factors, such as task contexts, examples, etc. We then proposed methods for automatically suggesting editing actions to improve task quality. Next, we studied two important task properties, i.e. complexity in Chapter 6 (based on [232]) and clarity in Chapter 7 (based on [72]), which are highly uncorrelated as we showed. From the point of view of crowds, we investigated how the perception of task complexity and clarity can be influenced by task design features, including metadata features (e.g. reward), the description of the content, and the visual design (e.g. colourfulness). We then proposed automatic methods to measure task complexity and clarity based on these task features. We thus contribute approaches to assist knowledge demanders in task design for improving crowd experience and task performance.

The third part of the thesis addresses the problem of task assignment to better associate tasks to crowds. We designed novel recommendation techniques that can fully exploit crowd and task properties for optimal crowd-task association. First, we investigated the effect of knowledge-related features of crowds and task topics on task recommendation in Chapter 8 (based on

[230]). We proposed a learning-to-rank method to account for both aspects for improving task recommendation. To further enhance recommendation performance, we proposed to take into consideration the structured nature of crowd and task properties, which are often organized in taxonomies. By analyzing multiple recommendation datasets, we showed that different relationships of crowds and tasks can be induced from their structured properties, including similarity in Chapter 9 (based on [233]), and complementarity & alternativity in Chapter 10 (based on [207]). By designing two novel recommendation methods, namely ReMF and HieVH, we showed that these relationships can not only help improve recommendation performance, but also boost the interpretability of recommendation results. Finally, we introduced MRLR in Chapter 11 (based on [208]) to capture crowd and task relationships in taxonomies by representation learning. Extensive experiments demonstrated that MRLR significantly outperformed state-of-the-art recommendation methods.

Samenvatting

Versnelling van Kenniscreatie met Menigten

Kenniscreatie met behulp van menigten speelt een centrale rol in veel typen van web-gebaseerde informatiesystemen, van zogenaamde "community question-answering (CQA)"-systemen (bijv. StackOverflow en Quora) tot zogenaamde "micro-task crowdsourcing"-systemen (bijv. Amazon mTurk en CrowdFlower). In deze systemen worden kennisbehoeften in het algemeen vervuld met behulp van taken (bijv. vragen in CQA-systemen of micro-taken in crowdsourcing-systemen) die worden uitgevoerd door een groep van individuen (bijv. deelnemers in CQA-systemen of werkers in crowdsourcing-systemen). Ondanks het succes in sommige platforms, werd tot nu toe van kenniscreatietaken aangenomen dat ze van lage cognitieve complexiteit zijn en in het algemeen in een bottom-up proces worden opgelost; als gevolg hangen de resultaten sterk af van de spontane en autonome bijdragen van menigten. Dit beperkt de mogelijkheid voor ons om het volume, de snelheid en de kwaliteit van de kenniscreatie te beheersen. Door kenmerken van menselijke kennis te benutten, bijv. expertise en motivatie, voorzien we dat kenniscreatie met menigten het volledige potentieel waarin complexe kennisintensieve taken worden opgelost en zo kennis van hoge kwaliteit efficiënt wordt gecreëerd.

Dit proefschrift richt zich daarom op het begrijpen van processen van kenniscreatie met menigten en het ontwikkelen van methoden en gereedschappen voor het beheersen en versnellen van de processen. Om voor dit doel de sleutelstappen in kenniscreatie met menigten te beschouwen voeren we de discussie aan de hand van een generiek model [231], dat bouwt op de volgende sleutelcomponenten: ***1)*** Technieken voor het *modelleren van menigten*, om kenmerken te beoordelen gerelateerd aan de kennis van menigten; ***2)*** Technieken voor het *modelleren van taken*, om kennisbehoeften en middelen voor kenniscreatie te representeren; ***3)*** Methoden voor het *toewijzen van taken*, om taken met menigten te associëren. Ons werk heeft als ambitie om aan

te tonen hoe, door het optimaal ontwerpen van elk van de boven beschreven componenten, het mogelijk is om kenniscreatie met menigten te versnellen op een principiële en effectieve manier.

Het eerste deel van dit proefschrift introduceert ons werk aan het modelleren van menigten. We dragen bij aan de wetenschap met een nieuw begrip van expertise en nieuwe metrieken voor expertise, een belangrijke en veelzijdige eigenschap van individuen en gemeenschappen. Geïnspireerd door theorieën over expertise in de sociologie, stellen we in hoofdstuk 2 (gebaseerd op [229]) een nieuwe expertisemetriek voor op basis van sociale oordelen, namelijk de "Mean Expertise Contribution" (MEC). Door deze te vergelijken met bestaande expertisemetrieken, hebben we empirisch vastgesteld dat MEC beter de expertise van menigten kan karakteriseren dan traditionele metrieken die bevooroordeeld zijn ten aanzien van activiteit en inzet. Om recht te doen aan de multidimensionale manifestatie van expertise in kenniscreatie, deling van middelen, en sociale interactie, hebben we daarna in Hoofdstuk 3 onze studie uitgebreid naar het karakteriseren van expertise over verschillende platformen heen. We hebben laten zien dat verschillende activiteiten van menigten over verschillende platformen heen kunnen helpen om verschillende kenmerken van expertise te karakteriseren, i.c. specialistische expertise en alomtegenwoordige expertise. Daarna hebben we activiteiten van menigten als gemeenschappen geanalyseerd en in Hoofdstuk 4 aangetoond hoe gemeenschapsactiviteiten invloed kunnen hebben op de dynamiek van marktplaatsen voor kenniscreatie en erdoor beïnvloed kunnen worden. Onze resultaten werpen een nieuw licht op het belang van sociale en werkgerelateerde voorkeuren van gemeenschappen voor de effectiviteit van taken.

Het tweede deel van het proefschrift richt zich op het modelleren van taken. We bestudeerden een verzameling eigenschappen van taken die verband houden met de kwaliteit en snelheid van de uitvoering van een taak door menigten, namelijk: de kwaliteit van de taakformulering, de complexiteit van de taak, en de helderheid van de taak. Gedreven door een groot aandeel van slecht geformuleerde taken, hebben we in Hoofdstuk 5 (gebaseerd op [228]) eerst factoren met betrekking tot de taakinhoud geanalyseerd die de taakkwaliteit substantieel zouden kunnen beïnvloeden. Een kwalitatieve studie leverde 7 belangrijke inhoudsfactoren, zoals de context van een taak, voorbeelden, etc. Daarna hebben we methoden voorgesteld voor het automatisch suggereren van redactieacties om de taakkwaliteit te verbeteren. Vervolgens hebben we twee belangrijke taakeigenschappen bestudeerd, namelijk complexiteit in Hoofdstuk 6 (gebaseerd op [232]) en helderheid in Hoofdstuk 7 (gebaseerd op [72]), twee eigenschappen die in hoge mate ongecorreleerd zijn zoals we aantoonden. Vanuit het gezichtspunt van menigten, onderzochten

we hoe de perceptie van taakcomplexiteit en –helderheid beïnvloed kan worden door kenmerken van het taakontwerp, inclusief metadata-kenmerken (bijv. beloning), de beschrijving van de content, en het visueel ontwerp (bijv. de mate van kleuring). We hebben daarna automatische methoden voorgesteld voor het meten van taakcomplexiteit en –helderheid op basis van deze taakkenmerken. Daarmee dragen we wetenschappelijk bij met aanpakken om kennisvragers te ondersteunen in het ontwerp van taken die de ervaring van de menigte en de effectiviteit van de taak verbeteren.

Het derde deel van het proefschrift beschouwt het probleem van de taaktoewijzing om taken beter aan menigtes te kunnen toewijzen. We ontwikkelden nieuwe aanbevelingstechnieken die volledig de eigenschappen van menigte en taak benutten voor een optimaal verband van menigten en taken. In de eerste plaats onderzochten we in Hoofdstuk 8 (gebaseerd op [230]) het effect van kennis-gerelateerde kenmerken van menigten en taakonderwerpen op de aanbeveling van taken. We stelden een zogenaamde "learning-to-rank"-methode voor om beide aspecten mee te kunnen nemen voor het verbeteren van taakaanbevelingen. Om het effect van de aanbevelingen nog verder te versterken hebben we voorgesteld om de gestructureerde aard te beschouwen van eigenschappen van menigte en taak; deze structuren zijn vaak georganiseerd in taxonomieën. Door meerdere datasets voor aanbevelingen te analyseren, hebben we laten zien dat verschillende relaties tussen menigten en taken afgeleid kunnen worden uit hun structurele eigenschappen, inclusief similariteit in Hoofdstuk 9 (gebaseerd op [233]) en complementariteit en alternativiteit in Hoofdstuk 10 (gebaseerd op [207]). Door het ontwerp van twee nieuwe aanbevelingsmethoden, namelijk ReMF en HieVH, toonden we aan dat deze relaties niet alleen kunnen helpen om de effecten van aanbevelingen te verbeteren, maar ook om de mogelijkheden flink te versterken om resultaten van aanbevelingen te interpreteren. Tenslotte introduceerden we in Hoofdstuk 11 (gebaseerd op [208]) MRLR om met behulp van "representation learning" relaties van menigten en taken in taxonomieën uit te drukken. Extensieve experimenten toonden aan dat MRLR significant beter presteert dan de "state of the art" in aanbevelingsmethoden.

Curriculum Vitae

Jie Yang was born in Wenzhou, China on Feb. 17, 1990. He received his master degree with cum laude from Eindhoven University of Technology, Netherlands. His master thesis was completed at Philips Research (Eindhoven), where he worked on the problem of time series classification. Before, he received his bachelor degree from Zhejiang University, China.

From Sep. 2013 to Sep. 2017, Jie Yang was a PhD student in the Web Information Systems group at Delft University of Technology, supervised by Geert-Jan Houben and Alessandro Bozzon. His PhD work focused on data-driven approaches for better understanding and accelerating crowd knowledge creation processes, by creating novel methods and tools for user modeling, recommendation, and crowdsourcing and human computation. During his PhD, he interned as a machine learning scientist in Amazon (Seattle), where he conducted research on deep active learning from crowds. Jie's research has been published in leading conferences and journals on related domains (e.g., HCOMP, UMAP, RecSys, IJCAI, AAAI, CIKM, HyperText, etc.). He received Best Paper Awards from ACM HyperText 2017 and from TRUSTINCW co-held with ACM WebSci 2016. Jie co-organized the International Workshop on Recommender Systems for Citizens (CitRec) at ACM RecSys 2017. He has also served as a program committee member and reviewer for several conferences and journals, such as WWW, HCOMP, IJCAI, UMAP, IUI, ICDM, TKDE, ECRA, JWE, etc.

Publications

1. Wenjie Pei*, Jie Yang*, Zhu Sun, Jie Zhang, Alessandro Bozzon, David Tax. Interacting Attention-Gated Recurrent Networks for Recommendation. In Proceedings of the 26th ACM International Conference on Information and Knowledge Management (CIKM 2017), full paper, Singpore, November 6 - 10, 2017, ACM. (*Joint first authors)

2. Jie Yang, Zhu Sun, Alessandro Bozzon, Jie Zhang, Martha Larson. CitRec 2017: International Workshop on Recommender Systems for Citizens. In Proceedings of the 11th ACM Conference on Recommender Systems (RecSys 2017), workshop summary paper, Commo, Italy, August 27 - 31, 2017, ACM.

3. Zhu Sun*, Jie Yang*, Jie Zhang, Alessandro Bozzon, Yu Chen, Chi Xu. MRLR: Multi-level Representation Learning for Personalized Ranking in Recommendation. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), full paper, Melbourne, Australia, August 19 - 25, 2017. (*Joint first authors)

4. Ujwal Gadiraju, Jie Yang, Alessandro Bozzon. Clarity is a Worthwhile Quality - On the Role of Task Clarity in Microtask Crowdsourcing. In Proceedings of the 28st ACM Conference on Hypertext and Social Media (HyperText 2017), full paper, Prague, Czech Republic, July 4 - 7, 2017, ACM.

5. Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon. Exploiting both Vertical and Horizontal Dimensions of Feature Hierarchy for Effective Recommendation. In Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017), full paper, San Francisco, California, US, February 4 - 9, p. 189-195, 2017, AAAI.

6. Jie Yang, Judith Redi, Alessandro Bozzon, Gianluca Demartini. Modeling Crowdsourcing Task Complexity. In Proceedings of the 4th AAAI Conference on Human Computation and Crowdsourcing (HCOMP 2016), full paper, Austin, TX, US, October 30 - November 3, p. 249-258, 2016, AAAI.

7. Arkka Dhiratara, Jie Yang, Alessandro Bozzon, Geert-Jan Houben. Social Media Data Analytics for Tourism – A Preliminary Study. In Proceedings of the 2nd Knowledge Discovery on the Web (KDWeb 2016), full paper, Cagliari, Italy, September 08 - 10, 2016.

8. Jie Yang, Zhu Sun, Alessandro Bozzon, Jie Zhang. Learning Hierarchical Feature Influence for Recommendation by Recursive Regularization. In Proceedings of the 10th ACM Conference on Recommender Systems (RecSys 2016), full paper, Boston, MA, US, September 15 - 19, p. 51-58, 2016, ACM.

9. Jie Yang, Alessandro Bozzon. On the Improvement of Quality and Reliability of Trust Cues in Micro-task Crowdsourcing. In the 1st ACM

Web Science Workshop on Weaving Relations of Trust in Crowd Work: Transparency and Reputation across Platforms. (TURSTINCW@WebSci 2016), position paper, Hannover, Germany, May 22 - 25, 2016, ACM.

10. Deniz Iren, Cynthia Liem, Jie Yang, Alessandro Bozzon. Using Social Media to Reveal Social and Collective Perspectives on Music. In Proceedings of the 8th ACM Conference on Web Science (WebSci 2016), short paper, Hannover, Germany, May 22 - 25, p. 296-300, 2016, ACM.

11. Jie Yang, Claudia Hauff, Geert-Jan Houben, Christiaan Titos Bolivar. Diversity in Social Media Urban Analytics.In Proceedings of the 16th International Conference on Web Engineering (ICWE 2016), full paper, Lugano, Switzerland, June 6 - 9, p. 335-353, 2016, Springer LNCS.

12. Yuan, Lu, Jie Yang. Notes on Low-rank Matrix Factorization. In arxiv:1507.00333, 2015.

13. Giuseppe Silvestri, Jie Yang, Alessandro Bozzon, Andrea Tagarelli. Linking Accounts across Social Networks: the Case of StackOverflow, Github and Twitter. In Proceedings of the 1st Knowledge Discovery on the Web (KDWeb 2015), full paper, Cagliari, Italy, September 03 - 05, 2015, p. 41-52, CEUR Vol-1489.

14. Jasper Oosterman, Jie Yang, Alessandro Bozzon, Lora Aroyo, Geert-Jan Houben. Crowdsourcing Visual Artwork Annotations in Cultural Heritage. In Computer Networks 2015, 90 133-149, Elsevier, 2015.

15. Jie Yang, Alessandro Bozzon, Geert-Jan Houben. Harnessing Engagement for Knowledge Creation Acceleration in Collaborative Q&A Systems. In Proceedings of the 23rd Conference on User Modeling, Adaption and Personalization (UMAP 2015), full paper, Dublin, Ireland, June 29 - July 03, p. 315-327, 2015, Springer LNC 9146.

16. Jie Yang, Alessandro Bozzon, Geert-Jan Houben. E-Wise: an Expertise-Driven Recommendation Platform for Web Question Answering Systems. In Proceedings of the 15th International Conference on Web Engineering (ICWE 2015), demonstration paper, Rotterdam, the Netherlands, June 23 - 26, 2015, p. 691-694, 2015, Springer LNCS 9114.

17. Jie Yang, Alessandro Bozzon, Geert-Jan Houben. Knowledge Crowdsourcing Acceleration. In Proceedings of the 15th International Conference on Web Engineering (ICWE 2015), doctoral symposium paper, Rotterdam, the Netherlands, June 23 - 26, 2015, p. 691-694, 2015, Springer LNCS 9114.

18. Jie Yang, Claudia Hauff, Alessandro Bozzon, Geert-Jan Houben. Asking the Right Question in Collaborative Q&A Systems. In Proceedings of the 25st ACM Conference on Hypertext and Social Media (HyperText 2014), full paper, Santiago Downtown, Chile, September 1 - 4, p. 179-189, 2014, ACM.

19. Jie Yang, Ke Tao, Alessandro Bozzon, Geert-Jan Houben. Sparrows and Owls: Characterisation of Expert Behaviour in StackOverflow. In Proceedings of the 22nd Conference on User Modeling, Adaption and Personalization (UMAP 2014), full paper, Aalborg, Denmark, July 7 - 11, 2014, p. 266-277, 2014, Springer LNCS.

20. Xi Long, Jie Yang, Tim Weysen, Reinder Haakma, Jerome Foussier, Pedro Fonseca, Ronald M. Aarts. Measuring Dissimilarity Between Respiratory Effort Signals based on Uniform Scaling for Sleep Staging. In Physiological Measurement 2014, 35 (12) 2529-2542, 2014, Institude of Physics and Engineering in Medicine.

21. Hoang Thanh Lam, Toon Calders, Jie Yang, Fabian Moerchen, Dmitriy Fradkin. Zips: Mining Compressing Sequential Patterns in Streams. In Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics (IDEA@KDD 2013), full paper, Chicago, Illinois, USA, August 11 - 14, 2013, p. 54-62, 2013, ACM.

SIKS Dissertation Series

Since 1998, all dissertations written by Ph.D. students who have conducted their research under auspices of a senior research fellow of the SIKS research school are published in the SIKS Dissertation Series.

2017-47 Jie Yang (TUD), *Crowd Knowledge Creation Acceleration*
2017-46 Jan Schneider (OU), *Sensor-based Learning Support*
2017-45 Bas Testerink (UU), *Decentralized Runtime Norm Enforcement*
2017-44 Garm Lucassen (UU), *Understanding User Stories - Computational Linguistics in Agile Requirements Engineering*
2017-43 Maaike de Boer (RUN), *Semantic Mapping in Video Retrieval*
2017-42 Elena Sokolova (RUN), *Causal discovery from mixed and missing data with applications on ADHD datasets*
2017-41 Adnan Manzoor (VUA), *Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle*
2017-40 Altaf Hussain Abro (VUA), *Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems"*
2017-39 Sara Ahmadi (RUN), *Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR*
2017-38 Alex Kayal (TUD), *Normative Social Applications*
2017-37 Alejandro Montes Garca (TUE), *WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy*
2017-36 Yuanhao Guo (UL), *Shape Analysis for Phenotype Characterisation from High-throughput Imaging*
2017-35 Martine de Vos (VU), *Interpreting natural science spreadsheets*
2017-34 Maren Scheffel (OUN), *The Evaluation Framework for Learning Analytics*
2017-33 Brigit van Loggem (OU), *Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity*
2017-32 Thaer Samar (RUN), *Access to and Retrievability of Content in Web Archives*
2017-31 Ben Ruijl (UL), *Advances in computational methods for QFT calculations*
2017-30 Wilma Latuny (UVT), *The Power of Facial Expressions*
2017-29 Adel Alhuraibi (UVT), *From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT*
2017-28 John Klein (VU), *Architecture Practices for Complex Contexts*
2017-27 Michiel Joosse (UT), *Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors*
2017-26 Merel Jung (UT), *Socially intelligent robots that understand and respond to human touch*
2017-25 Veruska Zamborlini (VU), *Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search*
2017-24 Chang Wang (TUD), *Use of Affordances for Efficient Robot Learning*
2017-23 David Graus (UVA), *Entities of Interest—Discovery in Digital Traces*
2017-22 Sara Magliacane (VU), *Logics for causal inference under uncertainty*
2017-21 Jeroen Linssen (UT), *Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)*
2017-20 Mohammadbashir Sedighi (TUD), *Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility*
2017-19 Jeroen Vuurens (TUD), *Proximity of Terms, Texts and Semantic Vectors in Information Retrieval*
2017-18 Ridho Reinanda (UVA), *Entity Associations for Search*
2017-17 Daniel Dimov (UL), *Crowdsourced Online Dispute Resolution*
2017-16 Aleksandr Chuklin (UVA), *Understanding and Modeling Users of Modern Search Engines*
2017-15 Peter Berck, Radboud University (RUN), *Memory-Based Text Correction*
2017-14 Shoshannah Tekofsky (UvT), *You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior*
2017-13 Gijs Huisman (UT), *Social Touch Technology - Extending the reach of social touch through haptic technology*
2017-12 Sander Leemans (TUE), *Robust Process Mining with Guarantees*
2017-11 Florian Kunneman (RUN), *Modelling patterns of time and emotion in Twitter #anticipointment*
2017-10 Robby van Delden (UT), *(Steering) Interactive Play Behavior*
2017-09 Dong Nguyen (UT), *Text as Social and Cultural Data: A Computational Perspective on Variation in Text*
2017-08 Rob Konijn (VU), *Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery*
2017-07 Roel Bertens (UU), *Insight in Information: from Abstract to Anomaly*
2017-06 Damir Vandic (EUR), *Intelligent Information Systems for Web Product Search*
2017-05 Mahdieh Shadi (UVA), *Collaboration Behavior*
2017-04 Mrunal Gawade (CWI), *MULTI-CORE PARALLELISM IN A COLUMN-STORE*
2017-03 Daniël Harold Telgen (UU), *Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines*

2017-02 Sjoerd Timmer (UU), *Designing and Understanding Forensic Bayesian Networks using Argumentation*
2017-01 Jan-Jaap Oerlemans (UL), *Investigating Cybercrime*
2016-50 Yan Wang (UVT), *The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains*
2016-49 Gleb Polevoy (TUD), *Participation and Interaction in Projects. A Game-Theoretic Analysis*
2016-48 Tanja Buttler (TUD), *Collecting Lessons Learned*
2016-47 Christina Weber (UL), *Real-time foresight - Preparedness for dynamic innovation networks*
2016-46 Jorge Gallego Perez (UT), *Robots to Make you Happy*
2016-45 Bram van de Laar (UT), *Experiencing Brain-Computer Interface Control*
2016-44 Thibault Sellam (UVA), *Automatic Assistants for Database Exploration*
2016-43 Saskia Koldijk (RUN), *Context-Aware Support for Stress Self-Management: From Theory to Practice*
2016-42 Spyros Martzoukos (UVA), *Combinatorial and Compositional Aspects of Bilingual Aligned Corpora*
2016-41 Thomas King (TUD), *Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance*
2016-40 Christian Detweiler (TUD), *Accounting for Values in Design*
2016-39 Merijn Bruijnes (UT), *Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect*
2016-38 Andrea Minuto (UT), *MATERIALS THAT MATTER - Smart Materials meet Art & Interaction Design*
2016-37 Giovanni Sileno (UvA), *Aligning Law and Action - a conceptual and computational inquiry*
2016-36 Daphne Karreman (UT), *Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies*
2016-35 Zhaochun Ren (UVA), *Monitoring Social Media: Summarization, Classification and Recommendation*
2016-34 Dennis Schunselaar (TUE), *Configurable Process Trees: Elicitation, Analysis, and Enactment*
2016-33 Peter Bloem (UVA), *Single Sample Statistics, exercises in learning from just one example*
2016-32 Eelco Vriezekolk (UT), *Assessing Telecommunication Service Availability Risks for Crisis Organisations*
2016-31 Mohammad Khelghati (UT), *Deep web content monitoring*
2016-30 Ruud Mattheij (UvT), *The Eyes Have It*
2016-29 Nicolas Hning (TUD), *Peak reduction in decentralised electricity systems -Markets and prices for flexible planning*
2016-28 Mingxin Zhang (TUD), *Large-scale Agent-based Social Simulation - A study on epidemic prediction and control*
2016-27 Wen Li (TUD), *Understanding Geo-spatial Information on Social Media*
2016-26 Dilhan Thilakarathne (VU), *In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains*
2016-25 Julia Kiseleva (TU/e), *Using Contextual Information to Understand Searching and Browsing Behavior*
2016-24 Brend Wanders (UT), *Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach*
2016-23 Fei Cai (UVA), *Query Auto Completion in Information Retrieval*
2016-22 Grace Lewis (VU), *Software Architecture Strategies for Cyber-Foraging Systems*
2016-21 Alejandro Moreno Clleri (UT), *From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground*
2016-20 Daan Odijk (UVA), *Context & Semantics in News & Web Search*
2016-19 Julia Efremova (Tu/e), *Mining Social Structures from Genealogical Data*
2016-18 Albert Meroo Peuela (VU), *Refining Statistical Data on the Web*
2016-17 Berend Weel (VU), *Towards Embodied Evolution of Robot Organisms*
2016-16 Guangliang Li (UVA), *Socially Intelligent Autonomous Agents that Learn from Human Reward*
2016-15 Steffen Michels (RUN), *Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments*
2016-14 Ravi Khadka (UU), *Revisiting Legacy Software System Modernization*
2016-13 Nana Baah Gyan (VU), *The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach*
2016-12 Max Knobbout (UU), *Logics for Modelling and Verifying Normative Multi-Agent Systems*
2016-11 Anne Schuth (UVA), *Search Engines that Learn from Their Users*
2016-10 George Karafotias (VUA), *Parameter Control for Evolutionary Algorithms*
2016-09 Archana Nottamkandath (VU), *Trusting Crowdsourced Information on Cultural Artefacts*
2016-08 Matje van de Camp (TiU), *A Link to the Past: Constructing Historical Social Networks from Unstructured Data*
2016-07 Jeroen de Man (VU), *Measuring and modeling negative emotions for virtual training*
2016-06 Michel Wilson (TUD), *Robust scheduling in an uncertain environment*
2016-05 Evgeny Sherkhonov (UVA), *Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers*
2016-04 Laurens Rietveld (VU), *Publishing and Consuming Linked Data*
2016-03 Maya Sappelli (RUN), *Knowledge Work in Context: User Centered Knowledge Worker Support*
2016-02 Michiel Christiaan Meulendijk (UU), *Optimizing medication reviews through decision support: prescribing a better pill to swallow*
2016-01 Syed Saiden Abbas (RUN), *Recognition of Shapes by Humans and Machines*
2015-35 Jungxao Xu (TUD), *Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction*
2015-34 Victor de Graaf (UT), *Gesocial Recommender Systems*
2015-33 Frederik Schadd (TUD), *Ontology Mapping with Auxiliary Resources*
2015-32 Jerome Gard (UL), *Corporate Venture Management in SMEs*
2015-31 Yakup Koç (TUD), *On the robustness of Power Grids*
2015-30 Kiavash Bahreini (OU), *Real-time Multimodal Emotion Recognition in E-Learning*
2015-29 Hendrik Baier (UM), *Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains*
2015-28 Janet Bagorogoza (TiU), *KNOWLEDGE MANAGEMENT AND HIGH PERFORMANCE; The Uganda Financial Institutions Model for HPO*
2015-27 Sándor Héman (CWI), *Updating compressed colomn stores*
2015-26 Alexander Hogenboom (EUR), *Sentiment Analysis of Text Guided by Semantics and Structure*
2015-25 Steven Woudenberg (UU), *Bayesian Tools for Early Disease Detection*
2015-24 Richard Berendsen (UVA), *Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation*
2015-23 Luit Gazendam (VU), *Cataloguer Support in Cultural Heritage*
2015-22 Zhemin Zhu (UT), *Co-occurrence Rate Networks*
2015-21 Sibren Fetter (OUN), *Using Peer-Support to Expand and Stabilize Online Learning*
2015-20 Loïs Vanhée (UU), *Using Culture and Values to Support Flexible Coordination*
2015-19 Bernardo Tabuenca (OUN), *Ubiquitous Technology for Lifelong Learners*
2015-18 Holger Pirk (CWI), *Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories*
2015-17 André van Cleeff (UT), *Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs*
2015-16 Changyun Wei (UT), *Cognitive Coordination for Cooperative Multi-Robot Teamwork*
2015-15 Klaas Andries de Graaf (VU), *Ontology-based Software Architecture Documentation*
2015-14 Bart van Straalen (UT), *A cognitive approach to modeling bad news conversations*

2015-13 Giuseppe Procaccianti (VU), *Energy-Efficient Software*
2015-12 Julie M. Birkholz (VU), *Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks*
2015-11 Yongming Luo (TUE), *Designing algorithms for big graph datasets: A study of computing bisimulation and joins*
2015-10 Henry Hermans (OUN), *OpenU: design of an integrated system to support lifelong learning*
2015-09 Randy Klaassen (UT), *HCI Perspectives on Behavior Change Support Systems*
2015-08 Jie Jiang (TUD), *Organizational Compliance: An agent-based model for designing and evaluating organizational interactions*
2015-07 Maria-Hendrike Peetz (UvA), *Time-Aware Online Reputation Analysis*
2015-06 Farideh Heidari (TUD), *Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes*
2015-05 Christoph Bösch (UT), *Cryptographically Enforced Search Pattern Hiding*
2015-04 Howard Spoelstra (OUN), *Collaborations in Open Learning Environments*
2015-03 Twan van Laarhoven (RUN), *Machine learning for network data*
2015-02 Faiza Bukhsh (UvT), *Smart auditing: Innovative Compliance Checking in Customs Controls*
2015-01 Niels Netten (UvA), *Machine Learning for Relevance of Information in Crisis Response*
2014-47 Shangsong Liang (UVA), *Fusion and Diversification in Information Retrieval*
2014-46 Ke Tao (TUD), *Social Web Data Analytics: Relevance, Redundancy, Diversity*
2014-45 Birgit Schmitz (OU), *Mobile Games for Learning: A Pattern-Based Approach*
2014-44 Paulien Meesters (UvT), *Intelligent Blauw. Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden*
2014-43 Kevin Vlaanderen (UU), *Supporting Process Improvement using Method Increments*
2014-42 Carsten Eickhoff (CWI/TUD), *Contextual Multidimensional Relevance Models*
2014-41 Frederik Hogenboom (EUR), *Automated Detection of Financial Events in News Text*
2014-40 Walter Oboma (RUN), *A Framework for Knowledge Management Using ICT in Higher Education*
2014-39 Jasmina Maric (UvT), *Web Communities, Immigration and Social Capital*
2014-38 Danny Plass-Oude Bos (UT), *Making brain-computer interfaces better: improving usability through post-processing*
2014-37 Maral Dadvar (UT), *Experts and Machines United Against Cyberbullying*
2014-36 Joos Buijs (TUE), *Flexible Evolutionary Algorithms for Mining Structured Process Models*
2014-35 Joost van Oijen (UU), *Cognitive Agents in Virtual Worlds: A Middleware Design Approach*
2014-34 Christina Manteli (VU), *The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems*
2014-33 Tesfa Tegegne Asfaw (RUN), *Service Discovery in eHealth*
2014-32 Naser Ayat (UVA), *On Entity Resolution in Probabilistic Data*
2014-31 Leo van Moergestel (UU), *Agent Technology in Agile Multiparallel Manufacturing and Product Support*
2014-30 Peter de Kock Berenschot (UvT), *Anticipating Criminal Behaviour*
2014-29 Jaap Kabbedijk (UU), *Variability in Multi-Tenant Enterprise Software*
2014-28 Anna Chmielowiec (VU), *Decentralized k-Clique Matching*
2014-27 Rui Jorge Almeida (EUR), *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*
2014-26 Tim Baarslag (TUD), *What to Bid and When to Stop*
2014-25 Martijn Lappenschaar (RUN), *New network models for the analysis of disease interaction*
2014-24 Davide Ceolin (VU), *Trusting Semi-structured Web Data*
2014-23 Eleftherios Sidirourgos (UvA/CWI), *Space Efficient Indexes for the Big Data Era*
2014-22 Marieke Peeters (UU), *Personalized Educational Games - Developing agent-supported scenario-based training*
2014-21 Kassidy Clark (TUD), *Negotiation and Monitoring in Open Environments*
2014-20 Mena Habib (UT), *Named Entity Extraction and Disambiguation for Informal Text: The Missing Link*
2014-19 Vincius Ramos (TUE), *Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support*
2014-18 Mattijs Ghijsen (VU), *Methods and Models for the Design and Study of Dynamic Agent Organizations*
2014-17 Kathrin Dentler (VU), *Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability*
2014-16 Krystyna Milian (VU), *Supporting trial recruitment and design by automatically interpreting eligibility criteria*
2014-15 Natalya Mogles (VU), *Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare*
2014-14 Yangyang Shi (TUD), *Language Models With Meta-information*
2014-13 Arlette van Wissen (VU), *Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains*
2014-12 Willem van Willigen (VU), *Look Ma, No Hands: Aspects of Autonomous Vehicle Control*
2014-11 Janneke van der Zwaan (TUD), *An Empathic Virtual Buddy for Social Support*
2014-10 Ivan Salvador Razo Zapata (VU), *Service Value Networks*
2014-09 Philip Jackson (UvT), *Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language*
2014-08 Samur Araujo (TUD), *Data Integration over Distributed and Heterogeneous Data Endpoints*
2014-07 Arya Adriansyah (TUE), *Aligning Observed and Modeled Behavior*
2014-06 Damian Tamburri (VU), *Supporting Networked Software Development*
2014-05 Jurriaan van Reijsen (UU), *Knowledge Perspectives on Advancing Dynamic Capability*
2014-04 Hanna Jochmann-Mannak (UT), *Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation*
2014-03 Sergio Raul Duarte Torres (UT), *Information Retrieval for Children: Search Behavior and Solutions*
2014-02 Fiona Tuliyano (RUN), *Combining System Dynamics with a Domain Modeling Method*
2014-01 Nicola Barile (UU), *Studies in Learning Monotone Models from Data*
2013-43 Marc Bron (UVA), *Exploration and Contextualization through Interaction and Concepts*
2013-42 Léon Planken (TUD), *Algorithms for Simple Temporal Reasoning*
2013-41 Jochem Liem (UVA), *Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning*
2013-40 Pim Nijssen (UM), *Monte-Carlo Tree Search for Multi-Player Games*
2013-39 Joop de Jong (TUD), *A Method for Enterprise Ontology based Design of Enterprise Information Systems*
2013-38 Eelco den Heijer (VU), *Autonomous Evolutionary Art*
2013-37 Dirk Börner (OUN), *Ambient Learning Displays*
2013-36 Than Lam Hoang (TUe), *Pattern Mining in Data Streams*
2013-35 Abdallah El Ali (UvA), *Minimal Mobile Human Computer Interaction*
2013-34 Kien Tjin-Kam-Jet (UT), *Distributed Deep Web Search*
2013-33 Qi Gao (TUD), *User Modeling and Personalization in the Microblogging Sphere*
2013-32 Kamakshi Rajagopal (OUN), *Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development*
2013-31 Dinh Khoa Nguyen (UvT), *Blueprint Model and Language for Engineering Cloud Applications*
2013-30 Joyce Nakatumba (TUE), *Resource-Aware Business Process Management: Analysis and Support*
2013-29 Iwan de Kok (UT), *Listening Heads*
2013-28 Frans van der Sluis (UT), *When Complexity becomes Interesting: An Inquiry into the Information eXperience*

2013-27 Mohammad Huq (UT), *Inference-based Framework Managing Data Provenance*
2013-26 Alireza Zarghami (UT), *Architectural Support for Dynamic Homecare Service Provisioning*
2013-25 Agnieszka Anna Latoszek-Berendsen (UM), *Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System*
2013-24 Haitham Bou Ammar (UM), *Automated Transfer in Reinforcement Learning*
2013-23 Patricio de Alencar Silva (UvT), *Value Activity Monitoring*
2013-22 Tom Claassen (RUN), *Causal Discovery and Logic*
2013-21 Sander Wubben (UvT), *Text-to-text generation by monolingual machine translation*
2013-20 Katja Hofmann (UvA), *Fast and Reliable Online Learning to Rank for Information Retrieval*
2013-19 Renze Steenhuizen (TUD), *Coordinated Multi-Agent Planning and Scheduling*
2013-18 Jeroen Janssens (UvT), *Outlier Selection and One-Class Classification*
2013-17 Koen Kok (VU), *The PowerMatcher: Smart Coordination for the Smart Electricity Grid*
2013-16 Eric Kok (UU), *Exploring the practical benefits of argumentation in multi-agent deliberation*
2013-15 Daniel Hennes (UM), *Multiagent Learning - Dynamic Games and Applications*
2013-14 Jafar Tanha (UVA), *Ensemble Approaches to Semi-Supervised Learning Learning*
2013-13 Mohammad Safiri (UT), *Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly*
2013-12 Marian Razavian (VU), *Knowledge-driven Migration to Services*
2013-11 Evangelos Pournaras (TUD), *Multi-level Reconfigurable Self-organization in Overlay Services*
2013-10 Jeewanie Jayasinghe Arachchige (UvT), *A Unified Modeling Framework for Service Design*
2013-09 Fabio Gori (RUN), *Metagenomic Data Analysis: Computational Methods and Applications*
2013-08 Robbert-Jan Merk (VU), *Making enemies: cognitive modeling for opponent agents in fighter pilot simulators*
2013-07 Giel van Lankveld (UvT), *Quantifying Individual Player Differences*
2013-06 Romulo Goncalves (CWI), *The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience*
2013-05 Dulce Pumareja (UT), *Groupware Requirements Evolutions Patterns*
2013-04 Chetan Yadati (TUD), *Coordinating autonomous planning and scheduling*
2013-03 Szymon Klarman (VU), *Reasoning with Contexts in Description Logics*
2013-02 Erietta Liarou (CWI), *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing*
2013-01 Viorel Milea (EUR), *News Analytics for Financial Decision Support*
2012-51 Jeroen de Jong (TUD), *Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching*
2012-50 Steven van Kervel (TUD), *Ontologogy driven Enterprise Information Systems Engineering*
2012-49 Michael Kaisers (UM), *Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions*
2012-48 Jorn Bakker (TUE), *Handling Abrupt Changes in Evolving Time-series Data*
2012-47 Manos Tsagkias (UVA), *Mining Social Media: Tracking Content and Predicting Behavior*
2012-46 Simon Carter (UVA), *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation*
2012-45 Benedikt Kratz (UvT), *A Model and Language for Business-aware Transactions*
2012-44 Anna Tordai (VU), *On Combining Alignment Techniques*
2012-42 Dominique Verpoorten (OU), *Reflection Amplifiers in self-regulated Learning*
2012-41 Sebastian Kelle (OU), *Game Design Patterns for Learning*
2012-40 Agus Gunawan (UvT), *Information Access for SMEs in Indonesia*
2012-39 Hassan Fatemi (UT), *Risk-aware design of value and coordination networks*
2012-38 Selmar Smit (VU), *Parameter Tuning and Scientific Testing in Evolutionary Algorithms*
2012-37 Agnes Nakakawa (RUN), *A Collaboration Process for Enterprise Architecture Creation*
2012-36 Denis Ssebugwawo (RUN), *Analysis and Evaluation of Collaborative Modeling Processes*
2012-35 Evert Haasdijk (VU), *Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics*
2012-34 Pavol Jancura (RUN), *Evolutionary analysis in PPI networks and applications*
2012-33 Rory Sie (OUN), *Coalitions in Cooperation Networks (COCOON)*
2012-32 Wietske Visser (TUD), *Qualitative multi-criteria preference representation and reasoning*
2012-31 Emily Bagarukayo (RUN), *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure*
2012-30 Alina Pommeranz (TUD), *Designing Human-Centered Systems for Reflective Decision Making*
2012-29 Almer Tigelaar (UT), *Peer-to-Peer Information Retrieval*
2012-28 Nancy Pascall (UvT), *Engendering Technology Empowering Women*
2012-27 Hayrettin Gurkok (UT), *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games*
2012-26 Emile de Maat (UVA), *Making Sense of Legal Text*
2012-25 Silja Eckartz (UT), *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application*
2012-24 Laurens van der Werff (UT), *Evaluation of Noisy Transcripts for Spoken Document Retrieval*
2012-23 Christian Muehl (UT), *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction*
2012-22 Thijs Vis (UvT), *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?*
2012-21 Roberto Cornacchia (TUD), *Querying Sparse Matrices for Information Retrieval*
2012-20 Ali Bahramisharif (RUN), *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing*
2012-19 Helen Schonenberg (TUE), *What's Next? Operational Support for Business Process Execution*
2012-18 Eltjo Poort (VU), *Improving Solution Architecting Practices*
2012-17 Amal Elgammal (UvT), *Towards a Comprehensive Framework for Business Process Compliance*
2012-16 Fiemke Both (VU), *Helping people by understanding them - Ambient Agents supporting task execution and depression treatment*
2012-15 Natalie van der Wal (VU), *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.*
2012-14 Evgeny Knutov (TUE), *Generic Adaptation Framework for Unifying Adaptive Web-based Systems*
2012-13 Suleman Shahid (UvT), *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions*
2012-12 Kees van der Sluijs (TUE), *Model Driven Design and Data Integration in Semantic Web Information Systems*
2012-11 J.C.B. Rantham Prabhakara (TUE), *Process Mining in the Large: Preprocessing, Discovery, and Diagnostics*
2012-10 David Smits (TUE), *Towards a Generic Distributed Adaptive Hypermedia Environment*
2012-09 Ricardo Neisse (UT), *Trust and Privacy Management Support for Context-Aware Service Platforms*
2012-08 Gerben de Vries (UVA), *Kernel Methods for Vessel Trajectories*
2012-07 Rianne van Lambalgen (VU), *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions*
2012-06 Wolfgang Reinhardt (OU), *Awareness Support for Knowledge Workers in Research Networks*
2012-05 Marijn Plomp (UU), *Maturing Interorganisational Information Systems*
2012-04 Jurriaan Souer (UU), *Development of Content Management System-based Web Applications*
2012-03 Adam Vanya (VU), *Supporting Architecture Evolution by Mining Software Repositories*

2012-02 Muhammad Umair (VU), *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models*
2012-01 Terry Kakeeto (UvT), *Relationship Marketing for SMEs in Uganda*
2011-49 Andreea Niculescu (UT), *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality*
2011-48 Mark Ter Maat (UT), *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent*
2011-47 Azizi Bin Ab Aziz (VU), *Exploring Computational Models for Intelligent Support of Persons with Depression*
2011-46 Beibei Hu (TUD), *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work*
2011-45 Herman Stehouwer (UvT), *Statistical Language Models for Alternative Sequence Selection*
2011-44 Boris Reuderink (UT), *Robust Brain-Computer Interfaces*
2011-43 Henk van der Schuur (UU), *Process Improvement through Software Operation Knowledge*
2011-42 Michal Sindlar (UU), *Explaining Behavior through Mental State Attribution*
2011-41 Luan Ibraimi (UT), *Cryptographically Enforced Distributed Data Access Control*
2011-40 Viktor Clerc (VU), *Architectural Knowledge Management in Global Software Development*
2011-39 Joost Westra (UU), *Organizing Adaptation using Agents in Serious Games*
2011-38 Nyree Lemmens (UM), *Bee-inspired Distributed Optimization*
2011-37 Adriana Burlutiu (RUN), *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference*
2011-36 Erik van der Spek (UU), *Experiments in serious game design: a cognitive approach*
2011-35 Maaike Harbers (UU), *Explaining Agent Behavior in Virtual Training*
2011-34 Paolo Turrini (UU), *Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations*
2011-33 Tom van der Weide (UU), *Arguing to Motivate Decisions*
2011-32 Nees-Jan van Eck (EUR), *Methodological Advances in Bibliometric Mapping of Science*
2011-31 Ludo Waltman (EUR), *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*
2011-30 Egon van den Broek (UT), *Affective Signal Processing (ASP): Unraveling the mystery of emotions*
2011-29 Faisal Kamiran (TUE), *Discrimination-aware Classification*
2011-28 Rianne Kaptein (UVA), *Effective Focused Retrieval by Exploiting Query Context and Document Structure*
2011-27 Aniel Bhulai (VU), *Dynamic website optimization through autonomous management of design patterns*
2011-26 Matthijs Aart Pontier (VU), *Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots*
2011-25 Syed Waqar ul Qounain Jaffry (VU)), *Analysis and Validation of Models for Trust Dynamics*
2011-24 Herwin van Welbergen (UT), *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior*
2011-23 Wouter Weerkamp (UVA), *Finding People and their Utterances in Social Media*
2011-22 Junte Zhang (UVA), *System Evaluation of Archival Description and Access*
2011-21 Linda Terlouw (TUD), *Modularization and Specification of Service-Oriented Systems*
2011-20 Qing Gu (VU), *Guiding service-oriented software engineering - A view-based approach*
2011-19 Ellen Rusman (OU), *The Mind's Eye on Personal Profiles*
2011-18 Mark Ponsen (UM), *Strategic Decision-Making in complex games*
2011-17 Jiyin He (UVA), *Exploring Topic Structure: Coherence, Diversity and Relatedness*
2011-16 Maarten Schadd (UM), *Selective Search in Games of Different Complexity*
2011-15 Marijn Koolen (UvA), *The Meaning of Structure: the Value of Link Evidence for Information Retrieval*
2011-14 Milan Lovric (EUR), *Behavioral Finance and Agent-Based Artificial Markets*
2011-13 Xiaoyu Mao (UvT), *Airport under Control. Multiagent Scheduling for Airport Ground Handling*
2011-12 Carmen Bratosin (TUE), *Grid Architecture for Distributed Process Mining*
2011-11 Dhaval Vyas (UT), *Designing for Awareness: An Experience-focused HCI Perspective*
2011-10 Bart Bogaert (UvT), *Cloud Content Contention*
2011-09 Tim de Jong (OU), *Contextualised Mobile Media for Learning*
2011-08 Nieske Vergunst (UU), *BDI-based Generation of Robust Task-Oriented Dialogues*
2011-07 Yujia Cao (UT), *Multimodal Information Presentation for High Load Human Computer Interaction*
2011-06 Yiwen Wang (TUE), *Semantically-Enhanced Recommendations in Cultural Heritage*
2011-05 Base van der Raadt (VU), *Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.*
2011-04 Hado van Hasselt (UU), *Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms*
2011-03 Jan Martijn van der Werf (TUE), *Compositional Design and Verification of Component-Based Information Systems*
2011-02 Nick Tinnemeier (UU), *Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language*
2011-01 Botond Cseke (RUN), *Variational Algorithms for Bayesian Inference in Latent Gaussian Models*
2010-53 Edgar Meij (UVA), *Combining Concepts and Language Models for Information Access*
2010-52 Peter-Paul van Maanen (VU), *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention*
2010-51 Alia Khairia Amin (CWI), *Understanding and supporting information seeking tasks in multiple sources*
2010-50 Bouke Huurnink (UVA), *Search in Audiovisual Broadcast Archives*
2010-49 Jahn-Takeshi Saito (UM), *Solving difficult game positions*
2010-47 Chen Li (UT), *Mining Process Model Variants: Challenges, Techniques, Examples*
2010-46 Vincent Pijpers (VU), *e3alignment: Exploring Inter-Organizational Business-ICT Alignment*
2010-45 Vasilios Andrikopoulos (UvT), *A theory and model for the evolution of software services*
2010-44 Pieter Bellekens (TUE), *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain*
2010-43 Peter van Kranenburg (UU), *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*
2010-42 Sybren de Kinderen (VU), *Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach*
2010-41 Guillaume Chaslot (UM), *Monte-Carlo Tree Search*
2010-40 Mark van Assem (VU), *Converting and Integrating Vocabularies for the Semantic Web*
2010-39 Ghazanfar Farooq Siddiqui (VU), *Integrative modeling of emotions in virtual agents*
2010-38 Dirk Fahland (TUE), *From Scenarios to components*
2010-37 Niels Lohmann (TUE), *Correctness of services and their composition*
2010-36 Jose Janssen (OU), *Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification*
2010-35 Dolf Trieschnigg (UT), *Proof of Concept: Concept-based Biomedical Information Retrieval*
2010-34 Teduh Dirgahayu (UT), *Interaction Design in Service Compositions*
2010-33 Robin Aly (UT), *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval*
2010-32 Marcel Hiel (UvT), *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems*
2010-31 Victor de Boer (UVA), *Ontology Enrichment from Heterogeneous Sources on the Web*

2010-30 Marieke van Erp (UvT), *Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval*
2010-29 Stratos Idreos (CWI), *Database Cracking: Towards Auto-tuning Database Kernels*
2010-28 Arne Koopman (UU), *Characteristic Relational Patterns*
2010-27 Marten Voulon (UL), *Automatisch contracteren*
2010-26 Ying Zhang (CWI), *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines*
2010-25 Zulfiqar Ali Memon (VU), *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective*
2010-24 Dmytro Tykhonov, *Designing Generic and Efficient Negotiation Strategies*
2010-23 Bas Steunebrink (UU), *The Logical Structure of Emotions*
2010-22 Michiel Hildebrand (CWI), *End-user Support for Access to Heterogeneous Linked Data*
2010-21 Harold van Heerde (UT), *Privacy-aware data management by means of data degradation*
2010-20 Ivo Swartjes (UT), *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*
2010-19 Henriette Cramer (UvA), *People's Responses to Autonomous and Adaptive Systems*
2010-18 Charlotte Gerritsen (VU), *Caught in the Act: Investigating Crime by Agent-Based Simulation*
2010-17 Spyros Kotoulas (VU), *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*
2010-16 Sicco Verwer (TUD), *Efficient Identification of Timed Automata, theory and practice*
2010-15 Lianne Bodenstaff (UT), *Managing Dependency Relations in Inter-Organizational Models*
2010-14 Sander van Splunter (VU), *Automated Web Service Reconfiguration*
2010-13 Gianluigi Folino (RUN), *High Performance Data Mining using Bio-inspired techniques*
2010-12 Susan van den Braak (UU), *Sensemaking software for crime analysis*
2010-11 Adriaan Ter Mors (TUD), *The world according to MARP: Multi-Agent Route Planning*
2010-10 Rebecca Ong (UL), *Mobile Communication and Protection of Children*
2010-09 Hugo Kielman (UL), *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging*
2010-08 Krzysztof Siewicz (UL), *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments*
2010-07 Wim Fikkert (UT), *Gesture interaction at a Distance*
2010-06 Sander Bakkes (UvT), *Rapid Adaptation of Video Game AI*
2010-05 Claudia Hauff (UT), *Predicting the Effectiveness of Queries and Retrieval Systems*
2010-04 Olga Kulyk (UT), *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*
2010-03 Joost Geurts (CWI), *A Document Engineering Model and Processing Framework for Multimedia documents*
2010-02 Ingo Wassink (UT), *Work flows in Life Science*
2010-01 Matthijs van Leeuwen (UU), *Patterns that Matter*
2009-46 Loredana Afanasiev (UvA), *Querying XML: Benchmarks and Recursion*
2009-45 Jilles Vreeken (UU), *Making Pattern Mining Useful*
2009-44 Roberto Santana Tapia (UT), *Assessing Business-IT Alignment in Networked Organizations*
2009-43 Virginia Nunes Leal Franqueira (UT), *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*
2009-42 Toine Bogers (UvT), *Recommender Systems for Social Bookmarking*
2009-41 Igor Berezhnyy (UvT), *Digital Analysis of Paintings*
2009-40 Stephan Raaijmakers (UvT), *Multinomial Language Learning: Investigations into the Geometry of Language*
2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin), *Service Substitution – A Behavioral Approach Based on Petri Nets*
2009-38 Riina Vuorikari (OU), *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*
2009-37 Hendrik Drachsler (OUN), *Navigation Support for Learners in Informal Learning Networks*
2009-36 Marco Kalz (OUN), *Placement Support for Learners in Learning Networks*
2009-35 Wouter Koelewijn (UL), *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling*
2009-34 Inge van de Weerd (UU), *Advancing in Software Product Management: An Incremental Method Engineering Approach*
2009-33 Khiet Truong (UT), *How Does Real Affect Affect Affect Recognition In Speech?*
2009-32 Rik Farenhorst (VU) and Remco de Boer (VU), *Architectural Knowledge Management: Supporting Architects and Auditors*
2009-31 Sofiya Katrenko (UVA), *A Closer Look at Learning Relations from Text*
2009-30 Marcin Zukowski (CWI), *Balancing vectorized query execution with bandwidth-optimized storage*
2009-29 Stanislav Pokraev (UT), *Model-Driven Semantic Integration of Service-Oriented Applications*
2009-28 Sander Evers (UT), *Sensor Data Management with Probabilistic Models*
2009-27 Christian Glahn (OU), *Contextual Support of social Engagement and Reflection on the Web*
2009-26 Fernando Koch (UU), *An Agent-Based Model for the Development of Intelligent Mobile Services*
2009-25 Alex van Ballegooij (CWI), *"RAM: Array Database Management through Relational Mapping"*
2009-24 Annerieke Heuvelink (VUA), *Cognitive Models for Training Simulations*
2009-23 Peter Hofgesang (VU), *Modelling Web Usage in a Changing Environment*
2009-22 Pavel Serdyukov (UT), *Search For Expertise: Going beyond direct evidence*
2009-21 Stijn Vanderlooy (UM), *Ranking and Reliable Classification*
2009-20 Bob van der Vecht (UU), *Adjustable Autonomy: Controling Influences on Decision Making*
2009-19 Valentin Robu (CWI), *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*
2009-18 Fabian Groffen (CWI), *Armada, An Evolving Database System*
2009-17 Laurens van der Maaten (UvT), *Feature Extraction from Visual Data*
2009-16 Fritz Reul (UvT), *New Architectures in Computer Chess*
2009-15 Rinke Hoekstra (UVA), *Ontology Representation - Design Patterns and Ontologies that Make Sense*
2009-14 Maksym Korotkiy (VU), *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*
2009-13 Steven de Jong (UM), *Fairness in Multi-Agent Systems*
2009-12 Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin), *Operating Guidelines for Services*
2009-11 Alexander Boer (UVA), *Legal Theory, Sources of Law & the Semantic Web*
2009-10 Jan Wielemaker (UVA), *Logic programming for knowledge-intensive interactive applications*
2009-09 Benjamin Kanagwa (RUN), *Design, Discovery and Construction of Service-oriented Systems*
2009-08 Volker Nannen (VU), *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*
2009-07 Ronald Poppe (UT), *Discriminative Vision-Based Recovery and Recognition of Human Motion*
2009-06 Muhammad Subianto (UU), *Understanding Classification*
2009-05 Sietse Overbeek (RUN), *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*
2009-04 Josephine Nabukenya (RUN), *Improving the Quality of Organisational Policy Making using Collaboration Engineering*
2009-03 Hans Stol (UvT), *A Framework for Evidence-based Policy Making Using IT*
2009-02 Willem Robert van Hage (VU), *Evaluating Ontology-Alignment Techniques*
2009-01 Rasa Jurgelenaite (RUN), *Symmetric Causal Independence Models*

2008-35 Ben Torben Nielsen (UvT), *Dendritic morphologies: function shapes structure*
2008-34 Jeroen de Knijf (UU), *Studies in Frequent Tree Mining*
2008-33 Frank Terpstra (UVA), *Scientific Workflow Design; theoretical and practical issues*
2008-32 Trung H. Bui (UT), *Toward Affective Dialogue Management using Partially Observable Markov Decision Processes*
2008-31 Loes Braun (UM), *Pro-Active Medical Information Retrieval*
2008-30 Wouter van Atteveldt (VU), *Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content*
2008-29 Dennis Reidsma (UT), *Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans*
2008-28 Ildiko Flesch (RUN), *On the Use of Independence Relations in Bayesian Networks*
2008-27 Hubert Vogten (OU), *Design and Implementation Strategies for IMS Learning Design*
2008-26 Marijn Huijbregts (UT), *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*
2008-25 Geert Jonker (UU), *Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency*
2008-24 Zharko Aleksovski (VU), *Using background knowledge in ontology matching*
2008-23 Stefan Visscher (UU), *Bayesian network models for the management of ventilator-associated pneumonia*
2008-22 Henk Koning (UU), *Communication of IT-Architecture*
2008-21 Krisztian Balog (UVA), *People Search in the Enterprise*
2008-20 Rex Arendsen (UVA), *Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven.*
2008-19 Henning Rode (UT), *From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search*
2008-18 Guido de Croon (UM), *Adaptive Active Vision*
2008-17 Martin Op 't Land (TUD), *Applying Architecture and Ontology to the Splitting and Allying of Enterprises*
2008-16 Henriette van Vugt (VU), *Embodied agents from a user's perspective*
2008-15 Martijn van Otterlo (UT), *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains.*
2008-14 Arthur van Bunningen (UT), *Context-Aware Querying; Better Answers with Less Effort*
2008-13 Caterina Carraciolo (UVA), *Topic Driven Access to Scientific Handbooks*
2008-12 Jozsef Farkas (RUN), *A Semiotically Oriented Cognitive Model of Knowledge Representation*
2008-11 Vera Kartseva (VU), *Designing Controls for Network Organizations: A Value-Based Approach*
2008-10 Wauter Bosma (UT), *Discourse oriented summarization*
2008-09 Christof van Nimwegen (UU), *The paradox of the guided user: assistance can be counter-effective*
2008-08 Janneke Bolt (UU), *Bayesian Networks: Aspects of Approximate Inference*
2008-07 Peter van Rosmalen (OU), *Supporting the tutor in the design and support of adaptive e-learning*
2008-06 Arjen Hommersom (RUN), *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*
2008-05 Bela Mutschler (UT), *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*
2008-04 Ander de Keijzer (UT), *Management of Uncertain Data - towards unattended integration*
2008-03 Vera Hollink (UVA), *Optimizing hierarchical menus: a usage-based approach*
2008-02 Alexei Sharpanskykh (VU), *On Computer-Aided Methods for Modeling and Analysis of Organizations*
2008-01 Katalin Boer-Sorbán (EUR), *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*
2007-25 Joost Schalken (VU), *Empirical Investigations in Software Process Improvement*
2007-24 Georgina Ramírez Camps (CWI), *Structural Features in XML Retrieval*
2007-23 Peter Barna (TUE), *Specification of Application Logic in Web Information Systems*
2007-22 Zlatko Zlatev (UT), *Goal-oriented design of value and process models from patterns*
2007-21 Karianne Vermaas (UU), *Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005*
2007-20 Slinger Jansen (UU), *Customer Configuration Updating in a Software Supply Network*
2007-19 David Levy (UM), *Intimate relationships with artificial partners*
2007-18 Bart Orriens (UvT), *On the development an management of adaptive business collaborations*
2007-17 Theodore Charitos (UU), *Reasoning with Dynamic Networks in Practice*
2007-16 Davide Grossi (UU), *Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems*
2007-15 Joyca Lacroix (UM), *NIM: a Situated Computational Memory Model*
2007-14 Niek Bergboer (UM), *Context-Based Image Analysis*
2007-13 Rutger Rienks (UT), *Meetings in Smart Environments; Implications of Progressing Technology*
2007-12 Marcel van Gerven (RUN), *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty*
2007-11 Natalia Stash (TUE), *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*
2007-10 Huib Aldewereld (UU), *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols*
2007-09 David Mobach (VU), *Agent-Based Mediated Service Negotiation*
2007-08 Mark Hoogendoorn (VU), *Modeling of Change in Multi-Agent Organizations*
2007-07 Natasa Jovanovic (UT), *To Whom It May Concern - Addressee Identification in Face-to-Face Meetings*
2007-06 Gilad Mishne (UVA), *Applied Text Analytics for Blogs*
2007-05 Bart Schermer (UL), *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance*
2007-04 Jurriaan van Diggelen (UU), *Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach*
2007-03 Peter Mika (VU), *Social Networks and the Semantic Web*
2007-02 Wouter Teepe (RUG), *Reconciling Information Exchange and Confidentiality: A Formal Approach*
2007-01 Kees Leune (UvT), *Access Control and Service-Oriented Architectures*
2006-28 Borkur Sigurbjornsson (UVA), *Focused Information Access using XML Element Retrieval*
2006-27 Stefano Bocconi (CWI), *Vox Populi: generating video documentaries from semantically annotated media repositories*
2006-26 Vojkan Mihajlovic (UT), *Score Region Algebra: A Flexible Framework for Structured Information Retrieval*
2006-25 Madalina Drugan (UU), *Conditional log-likelihood MDL and Evolutionary MCMC*
2006-24 Laura Hollink (VU), *Semantic Annotation for Retrieval of Visual Resources*
2006-23 Ion Juvina (UU), *Development of Cognitive Model for Navigating on the Web*
2006-22 Paul de Vrieze (RUN), *Fundaments of Adaptive Personalisation*
2006-21 Bas van Gils (RUN), *Aptness on the Web*
2006-20 Marina Velikova (UvT), *Monotone models for prediction in data mining*
2006-19 Birna van Riemsdijk (UU), *Cognitive Agent Programming: A Semantic Approach*
2006-18 Valentin Zhizhkun (UVA), *Graph transformation for Natural Language Processing*
2006-17 Stacey Nagata (UU), *User Assistance for Multitasking with Interruptions on a Mobile Device*
2006-16 Carsten Riggelsen (UU), *Approximation Methods for Efficient Learning of Bayesian Networks*
2006-15 Rainer Malik (UU), *CONAN: Text Mining in the Biomedical Domain*

2006-14 Johan Hoorn (VU), *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*
2006-13 Henk-Jan Lebbink (UU), *Dialogue and Decision Games for Information Exchanging Agents*
2006-12 Bert Bongers (VU), *Interactivation - Towards an e-cology of people, our technological environment, and the arts*
2006-11 Joeri van Ruth (UT), *Flattening Queries over Nested Data Types*
2006-10 Ronny Siebes (VU), *Semantic Routing in Peer-to-Peer Systems*
2006-09 Mohamed Wahdan (UM), *Automatic Formulation of the Auditor's Opinion*
2006-08 Eelco Herder (UT), *Forward, Back and Home Again - Analyzing User Behavior on the Web*
2006-07 Marko Smiljanic (UT), *XML schema matching – balancing efficiency and effectiveness by means of clustering*
2006-06 Ziv Baida (VU), *Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling*
2006-05 Cees Pierik (UU), *Validation Techniques for Object-Oriented Proof Outlines*
2006-04 Marta Sabou (VU), *Building Web Service Ontologies*
2006-03 Noor Christoph (UVA), *The role of metacognitive skills in learning to solve problems*
2006-02 Cristina Chisalita (VU), *Contextual issues in the design and use of information technology in organizations*
2006-01 Samuil Angelov (TUE), *Foundations of B2B Electronic Contracting*
2005-21 Wijnand Derks (UT), *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*
2005-20 Cristina Coteanu (UL), *Cyber Consumer Law, State of the Art and Perspectives*
2005-19 Michel van Dartel (UM), *Situated Representation*
2005-18 Danielle Sent (UU), *Test-selection strategies for probabilistic networks*
2005-17 Boris Shishkov (TUD), *Software Specification Based on Re-usable Business Components*
2005-16 Joris Graaumans (UU), *Usability of XML Query Languages*
2005-15 Tibor Bosse (VU), *Analysis of the Dynamics of Cognitive Processes*
2005-14 Borys Omelayenko (VU), *Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics*
2005-13 Fred Hamburg (UL), *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*
2005-12 Csaba Boer (EUR), *Distributed Simulation in Industry*
2005-11 Elth Ogston (VU), *Agent Based Matchmaking and Clustering - A Decentralized Approach to Search*
2005-10 Anders Bouwer (UVA), *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*
2005-09 Jeen Broekstra (VU), *Storage, Querying and Inferencing for Semantic Web Languages*
2005-08 Richard Vdovjak (TUE), *A Model-driven Approach for Building Distributed Ontology-based Web Applications*
2005-07 Flavius Frasincar (TUE), *Hypermedia Presentation Generation for Semantic Web Information Systems*
2005-06 Pieter Spronck (UM), *Adaptive Game AI*
2005-05 Gabriel Infante-Lopez (UVA), *Two-Level Probabilistic Grammars for Natural Language Parsing*
2005-04 Nirvana Meratnia (UT), *Towards Database Support for Moving Object data*
2005-03 Franc Grootjen (RUN), *A Pragmatic Approach to the Conceptualisation of Language*
2005-02 Erik van der Werf (UM)), *AI techniques for the game of Go*
2005-01 Floor Verdenius (UVA), *Methodological Aspects of Designing Induction-Based Applications*
2004-20 Madelon Evers (Nyenrode), *Learning from Design: facilitating multidisciplinary design teams*
2004-19 Thijs Westerveld (UT), *Using generative probabilistic models for multimedia retrieval*
2004-18 Vania Bessa Machado (UvA), *Supporting the Construction of Qualitative Knowledge Models*
2004-17 Mark Winands (UM), *Informed Search in Complex Games*
2004-16 Federico Divina (VU), *Hybrid Genetic Relational Search for Inductive Learning*
2004-15 Arno Knobbe (UU), *Multi-Relational Data Mining*
2004-14 Paul Harrenstein (UU), *Logic in Conflict. Logical Explorations in Strategic Equilibrium*
2004-13 Wojciech Jamroga (UT), *Using Multiple Models of Reality: On Agents who Know how to Play*
2004-12 The Duy Bui (UT), *Creating emotions and facial expressions for embodied agents*
2004-11 Michel Klein (VU), *Change Management for Distributed Ontologies*
2004-10 Suzanne Kabel (UVA), *Knowledge-rich indexing of learning-objects*
2004-09 Martin Caminada (VU), *For the Sake of the Argument; explorations into argument-based reasoning*
2004-08 Joop Verbeek (UM), *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise*
2004-07 Elise Boltjes (UM), *Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*
2004-06 Bart-Jan Hommes (TUD), *The Evaluation of Business Process Modeling Techniques*
2004-05 Viara Popova (EUR), *Knowledge discovery and monotonicity*
2004-04 Chris van Aart (UVA), *Organizational Principles for Multi-Agent Architectures*
2004-03 Perry Groot (VU), *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*
2004-02 Lai Xu (UvT), *Monitoring Multi-party Contracts for E-business*
2004-01 Virginia Dignum (UU), *A Model for Organizational Interaction: Based on Agents, Founded in Logic*
2003-18 Levente Kocsis (UM), *Learning Search Decisions*
2003-17 David Jansen (UT), *Extensions of Statecharts with Probability, Time, and Stochastic Timing*
2003-16 Menzo Windhouwer (CWI), *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses*
2003-15 Mathijs de Weerdt (TUD), *Plan Merging in Multi-Agent Systems*
2003-14 Stijn Hoppenbrouwers (KUN), *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*
2003-13 Jeroen Donkers (UM), *Nosce Hostem - Searching with Opponent Models*
2003-12 Roeland Ordelman (UT), *Dutch speech recognition in multimedia information retrieval*
2003-11 Simon Keizer (UT), *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*
2003-10 Andreas Lincke (UvT), *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*
2003-09 Rens Kortmann (UM), *The resolution of visually guided behaviour*
2003-08 Yongping Ran (UM), *Repair Based Scheduling*
2003-07 Machiel Jansen (UvA), *Formal Explorations of Knowledge Intensive Tasks*
2003-06 Boris van Schooten (UT), *Development and specification of virtual environments*
2003-05 Jos Lehmann (UVA), *Causation in Artificial Intelligence and Law - A modelling approach*
2003-04 Milan Petkovic (UT), *Content-Based Video Retrieval Supported by Database Technology*
2003-03 Martijn Schuemie (TUD), *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*
2003-02 Jan Broersen (VU), *Modal Action Logics for Reasoning About Reactive Systems*
2003-01 Heiner Stuckenschmidt (VU), *Ontology-Based Information Sharing in Weakly Structured Environments*
2002-17 Stefan Manegold (UVA), *Understanding, Modeling, and Improving Main-Memory Database Performance*
2002-16 Pieter van Langen (VU), *The Anatomy of Design: Foundations, Models and Applications*
2002-15 Rik Eshuis (UT), *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*
2002-14 Wieke de Vries (UU), *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*

2002-13 Hongjing Wu (TUE), *A Reference Architecture for Adaptive Hypermedia Applications*
2002-12 Albrecht Schmidt (Uva), *Processing XML in Database Systems*
2002-11 Wouter C.A. Wijngaards (VU), *Agent Based Modelling of Dynamics: Biological and Organisational Applications*
2002-10 Brian Sheppard (UM), *Towards Perfect Play of Scrabble*
2002-09 Willem-Jan van den Heuvel (KUB), *Integrating Modern Business Applications with Objectified Legacy Systems*
2002-08 Jaap Gordijn (VU), *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*
2002-07 Peter Boncz (CWI), *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*
2002-06 Laurens Mommers (UL), *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*
2002-05 Radu Serban (VU), *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*
2002-04 Juan Roberto Castelo Valdueza (UU), *The Discrete Acyclic Digraph Markov Model in Data Mining*
2002-03 Henk Ernst Blok (UT), *Database Optimization Aspects for Information Retrieval*
2002-02 Roelof van Zwol (UT), *Modelling and searching web-based document collections*
2002-01 Nico Lassing (VU), *Architecture-Level Modifiability Analysis*
2001-11 Tom M. van Engers (VUA), *Knowledge Management: The Role of Mental Models in Business Systems Design*
2001-10 Maarten Sierhuis (UvA), *Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design*
2001-09 Pieter Jan 't Hoen (RUL), *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*
2001-08 Pascal van Eck (VU), *A Compositional Semantic Structure for Multi-Agent Systems Dynamics.*
2001-07 Bastiaan Schonhage (VU), *Diva: Architectural Perspectives on Information Visualization*
2001-06 Martijn van Welie (VU), *Task-based User Interface Design*
2001-05 Jacco van Ossenbruggen (VU), *Processing Structured Hypermedia: A Matter of Style*
2001-04 Evgueni Smirnov (UM), *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*
2001-03 Maarten van Someren (UvA), *Learning as problem solving*
2001-02 Koen Hindriks (UU), *Agent Programming Languages: Programming with Mental Models*
2001-01 Silja Renooij (UU), *Qualitative Approaches to Quantifying Probabilistic Networks*
2000-11 Jonas Karlsson (CWI), *Scalable Distributed Data Structures for Database Management*
2000-10 Niels Nes (CWI), *Image Database Management System Design Considerations, Algorithms and Architecture*
2000-09 Florian Waas (CWI), *Principles of Probabilistic Query Optimization*
2000-08 Veerle Coupé (EUR), *Sensitivity Analyis of Decision-Theoretic Networks*
2000-07 Niels Peek (UU), *Decision-theoretic Planning of Clinical Patient Management*
2000-06 Rogier van Eijk (UU), *Programming Languages for Agent Communication*
2000-05 Ruud van der Pol (UM), *Knowledge-based Query Formulation in Information Retrieval.*
2000-04 Geert de Haan (VU), *ETAG, A Formal Model of Competence Knowledge for User Interface Design*
2000-03 Carolien M.T. Metselaar (UVA), *Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief.*
2000-02 Koen Holtman (TUE), *Prototyping of CMS Storage Management*
2000-01 Frank Niessink (VU), *Perspectives on Improving Software Maintenance*
1999-08 Jacques H.J. Lenting (UM), *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation.*
1999-07 David Spelt (UT), *Verification support for object database design*
1999-06 Niek J.E. Wijngaards (VU), *Re-design of compositional systems*
1999-05 Aldo de Moor (KUB), *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*
1999-04 Jacques Penders (UM), *The practical Art of Moving Physical Objects*
1999-03 Don Beal (UM), *The Nature of Minimax Search*
1999-02 Rob Potharst (EUR), *Classification using decision trees and neural nets*
1999-01 Mark Sloof (VU), *Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products*
1998-05 E.W.Oskamp (RUL), *Computerondersteuning bij Straftoemeting*
1998-04 Dennis Breuker (UM), *Memory versus Search in Games*
1998-03 Ans Steuten (TUD), *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*
1998-02 Floris Wiesman (UM), *Information Retrieval by Graphically Browsing Meta-Information*
1998-01 Johan van den Akker (CWI), *DEGAS - An Active, Temporal Database of Autonomous Objects*