

ASK THE RIGHT EXPERT

*QUESTION ROUTING BASED ON USER EXPERTISE IN WEB
QUESTIONS ANSWERING SYSTEMS*

Sijmen Peter Hoogendijk

ASK THE RIGHT EXPERT

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE
TRACK SOFTWARE TECHNOLOGY

by

Sijmen Peter Hoogendijk
born in Hoorn



Web Information Systems
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
<http://wis.ewi.tudelft.nl>

ASK THE RIGHT EXPERT

Author: Sijmen Peter Hoogendijk
Student id: 1379046
Email: s.p.hoogendijk@student.tudelft.nl

Abstract

Question Routing systems aim at routing questions to users that are more suited to answer them. Different techniques are used to match candidate users to questions, by considering properties of both. Existing techniques however do not consider the expertise of the candidate.

This work proposes an approach to Question Routing in which the user expertise is considered for question routing purposes. The proposed approach is a three stage process which allows for different configurations of existing matching techniques and user expertise. An experiment is set up in order to compare different Question Routing configurations. In total thirteen different configurations are evaluated, all based on three different content-based baseline methods. Stack Overflow is used as the source for questions, answers and users for the evaluation of the performance of different Question Routing configurations. A dataset containing 6 months worth of questions is used for the evaluation. The results show that incorporating expertise into Question Routing algorithms can provide significant performance increase.

Thesis Committee:

Chair:	Prof. dr. ir. G.J.P.M. Houben, Faculty EEMCS, TUDelft
University supervisor:	Dr. ir. A. Bozzon, Faculty EEMCS, TUDelft
Committee Member:	Dr. ir. A. Bacchelli, Faculty EEMCS, TUDelft
Committee Member:	Ir. J. Yang, Faculty EEMCS, TUDelft

Preface

Seven years before the writing of these words was my first day at Delft University of Technology. In those seven years I've had some ups and downs, I've made new friends and have learned many new skills. During those years I've had interest in many different parts of Computer Science. I started with an interest for programming, which gradually changed to an interest in computer graphics and artificial intelligence. After I noticed how hard it is to create a maintainable program my interest drifted towards software engineering. However since the course on Web and Semantic Web Engineering by Geert-Jan Houben, my interest for this field was sparked, especially for User Profiling.

I decided to do my Master thesis at Web Information Systems. It took a few weeks of figuring out what I wanted as the topic, but eventually I found one. Question recommendation for Stack Overflow.

In the year that followed, I've gained much more hands-on experience on text-processing, user modeling and user profiling. I've also learned (yet another) new programming language, just for the project. Even though I've learned a lot, I could not have completed this thesis without help. Therefore I would like to thank Alessandro Bozzon for guiding me through the process and helping out when I was stuck. I would like to thank Jie Yang for his support, his suggestions and ideas. I would like to thank everyone that provided comments and suggestions during our two-weekly Omicron meeting. Especially I would like to thank Claudia Hauff for some excellent critical questions that made me rethink and sometimes change my work.

Additionally, I would like to thank Geert-Jan Houben for inspiring me to do my thesis in this field and for being a member of my thesis committee. I would also like to thank Alberto Bacchelli for being a member of my thesis committee.

Finally I would like to thank my family and friends for supporting me, distracting me and encouraging me during the entire period that I stayed at the Delft University of Technology.

Sijmen Peter Hoogendijk
Delft, the Netherlands
December 4, 2014

Contents

Preface	iii
Contents	v
List of Figures	vii
1 Introduction	1
1.1 Motivation	2
1.2 Contribution	3
1.3 Document structure	3
2 Related work	5
2.1 Recommender systems	5
2.2 Question Routing in CQA	9
2.3 Modeling Users and Questions for Question Routing in CQA	10
2.4 Evaluating Question Routing Systems	12
2.5 Pitfalls of performance metrics	16
2.6 User expertise	16
3 Question Routing in CQA Systems	19
3.1 Common Data Model for CQA systems	19
3.2 Framework for Developing QR Systems	23
3.3 Preparation Stage	24
3.4 Matching Stage	25
3.5 Ranking Stage	28
3.6 Evaluation	29
4 Experimental Evaluation: Question Routing on Stack Overflow	35
4.1 Stack Overflow	35
4.2 Experimental Settings	36
4.3 Experimental Results	42
4.4 Discussion	50
5 Conclusions and Future Work	53

5.1	Conclusions	53
5.2	Discussion/Reflection	54
5.3	Future work	54
	Bibliography	57
A	Glossary	61
A.1	Terms	61
A.2	Abbreviations	61
B	Tables	63
C	Results for set $\mathcal{A}$	67
D	Results for set $\mathcal{B}$	75

List of Figures

1.1	A question with five answers on the CQA Yahoo! Answers.	1
2.1	Workflow of CBF Recommendation	6
2.2	Example of user based Collaborative Filtering	7
2.3	Example of item based Collaborative Filtering	8
3.1	A representation of the various parts of the CQA system Stack Overflow .	19
3.2	Simplified Data Model for CQA Systems.	20
3.3	The pipeline for creating various QR systems	24
3.4	The pipeline for processing text.	26
3.5	A time-line indicating which question and answers are taken as a subset. .	30
3.6	Venn diagram with Active User Questions and Debatable Questions . . .	31
3.7	Dividing a time period in 6 interaction slots.	32
3.8	An example of a valid 4/6 partitioning for active users.	32
4.1	An overview of the experimental flow.	36
4.2	User and answer distribution of the different datasets	39
4.3	Performance for different α 's	42
4.4	Results for NDCG for a 12/12 intensity for a six month period, set $\mathcal{B}$. . .	43
4.5	The performance on NDCG@5 for the content-based matching strategies. .	45
4.6	Performance of different expertise ranking strategies.	46
4.7	The performance of MEC compared to the best performing baseline. . . .	47
4.8	The two dimensions of testing in a single figure	48
C.1	Performance for different α 's	67
C.2	Results for NDCG for a 12/12 intensity for a three month period, set $\mathcal{A}$. .	68
C.3	The performance on NDCG@5 for the content-based matching strategies, set $\mathcal{A}$	68
C.4	Performance of different expertise ranking strategies, set $\mathcal{A}$	69
C.5	The performance of MEC compared to the best performing baseline, set $\mathcal{A}$	69
C.6	The two dimensions of testing in a single figure, set $\mathcal{A}$	70
D.1	Results for NDCG for a 12/12 intensity for a six month period, set $\mathcal{B}$. . .	76

D.2	The performance on NDCG@5 for the content-based matching strategies, set $\mathcal{B}$	76
D.3	Performance of different expertise ranking strategies, set $\mathcal{B}$	77
D.4	The performance of MEC compared to the best performing baseline, set $\mathcal{B}$	77
D.5	The two dimensions of testing in a single figure, set $\mathcal{B}$	78

Chapter 1

Introduction

In recent years there has been an increase in Community Question and Answering platforms (CQA) like Bǎidù zhidào¹, Yahoo! Answers² and Stack Exchange³. With the emerge of these platforms, the relation between askers and answerer has changed drastically. Previously questions would be asked to a person directly, or perhaps to multiple persons at once. In a CQA system questions are placed on the platform and can be answered by anyone who has access to the platform and thus are not directed to a single person. An example of a question posted on Yahoo! Answers is given in Figure 1.1, where there are five answerers for the question.

The advantage of Community Question and Answering platforms is that knowledge is created and distributed. Users that were previously unable to find a person that could answer their question now have a wealth of information available to find the answer. In the case an answer is not found by traditional means, a CQA platform offers the asker a much wider audience to ask their question.

However the undirected question asking has put a burden on the answerer who now must choose from the multitude of questions available in the platform [11]. This burden on the answerer has two consequences. First of all, there will be questions that remain unanswered, as there are typically a lot of questions but so few answerers available. Second the quality of the answers may suffer, as some answerers want to maximize the number of answered ques-

Science & Mathematics > Engineering Next ▶

Can anyone please tell me from where i can get a Schottky Diode ? ★

Update : like i cant purchase it but i wish to know that from where can i take it out from(i.e. appliances and stuff) . PLEASE HELP

Best Answer Asker's Choice

SMDave answered 2 days ago

Any switched mode power supply (such as for an iPhone) contains Schottky diodes. You can remove them with a desoldering tool or a flic chip removal kit.

Asker's rating & comment
★★★★

1 0 Comment

Other Answers (4) Relevance ▼

Steven answered 1 day ago

The BE of a silicon transistor has a (reverse) zener voltage about 7 volts. In series with the CB junction, the result is quit thermally stable. Remember the device has thermal / power limits regardless how you use it. A Zener diode must always be used with a resistor or other current limiting device. For example, a small 0.5W , 5V zener diode will burn up if you allow more than $0.5/5 = 0.1$ Amps through it.

0 0 Comment

lee28loo, 異地強索後人 answered 2 days ago

Any electronic store that sells parts !

1 0 Comment

2 answers hidden

Figure 1.1: A question with five answers on the CQA Yahoo! Answers. Two of the answers are hidden.

¹<http://zhidao.baidu.com/>

²<https://answers.yahoo.com/>

³<https://stackexchange.com/>

tions, and not the correctness and soundness of the explanation. Both cases degrade the effectiveness of the knowledge generation in the CQAs, which is their primary goal.

1.1 Motivation

To help creating a better knowledge base, the burden on the answerer should be reduced. One way of doing this is to suggest questions to candidate answerers such that the likelihood of obtaining a good answer is maximized. This process is called Question Routing (QR).

When a question enters a Question Routing system, the system predicts which users are most likely to give a good answer to the given question. To achieve this goal the system creates a profile for all users, which leverages information such as the number of answers given by the user and the users' preference for certain tags, to estimate how likely a user will answer a given question.

Almost all approaches in Question Routing consider the question content and user preferences for the question and user profile. Using only those features ignores one crucial aspect that exists in non-CQA question and answering; the knowledge level of the answerer, which might be reflected by the quality of a user's answers.

Current approaches do not consider the quality of answers provided by a user. Both a novice and an expert may both have the same preference, but the level of knowledge is quite different.

The evaluation of the knowledge level, or expertise, is currently not considered in most Question Routing approaches. Considering expertise is a natural extension of the currently existing approaches. Expertise will add extra information in the Question Routing system which would allow it to make a more informed decision. The working hypotheses is that including information about candidate answerer expertise can improve the performance of QR systems. The purpose of this work can be summarized in the following research question.

RQ How can user expertise improve the quality of Question Routing in a Community Question and Answering platform?

To answer this research question, several subquestions are formulated.

RQ1 How are existing Question Routing systems evaluated on performance?

RQ2 Which measures of expertise exist for Online Community Question and Answering Platforms?

RQ3 How to incorporate measures of expertise into Question Routing Systems?

RQ4 How do different measures of expertise influence the performance of Question Routing systems for Online Community Question and Answering?

In order to answer the research questions, several steps are taken. First a literature review is done in order to obtain information on how Question Routing systems are evaluated, and on how user expertise is measured in Online Community Question and Answering platforms. Second a reference process is proposed in order to facilitate

Question Routing system designs that incorporate expertise. Finally a extensive evaluation with four baseline Question Routing configurations and nine different Question Routing configurations including expertise is done on a dataset of an online reference CQA.

1.2 Contribution

This work features the following contributions:

1. An analysis of the evaluation techniques used to assess Question Routing systems, and the pitfalls of such techniques.
2. A reference process for designing Question Routing systems. This reference process is used to incorporate user expertise into Question Routing systems
3. An analysis of a reference online CQA. This provides insights into the unique problems that exist for Question Routing in the reference CQA platform.
4. Thirteen different Question Routing configurations are evaluated in order to provide insights into the effect of expertise on Question Routing systems.

1.3 Document structure

This document begins with an introduction in Chapter 1. In this introduction the problem description and relevance are given. The research questions are also proposed as well as the contributions of this work. Chapter 2 provides a in depth explanation of both Question Routing and Recommendation Systems and how both fields are related. In the same chapter methods for evaluating Question Routing systems are covered as well as the different models used in Question Routing. The chapter ends with an explanation of different measures of expertise. Chapter 3 introduces the reference process for creating Question Routing systems and several Question Routing systems are constructed using the proposed reference process. Chapter 4 introduces the experimental dimensions and provides and analyzes the results. The chapter finishes with a discussion of the results and the threat to validity. Finally the work is concluded in Chapter 5.

Chapter 2

Related work

Recommender Systems form the basis for Question Routing. The techniques used in both fields are very similar, but their purpose is different. In Recommender Systems the focus is on recommending items to users to aid the user in his quest for new and interesting content. In Question Routing the focus is on finding users that are able to answer a given question.

In this chapter the background and related work will be presented. In first Section different strategies for Recommender Systems are presented. The section that follows covers which recommender strategy is the best to use for Question Routing. Section 2.3 explains the different user and question models used in Question Routing. Next, Section 2.4 provides different metrics for evaluating the performance of a QR system. Since several metrics are common in both Question Routing and Recommender systems, Section 2.5 explains several differences between the results of Recommender Systems and Question Routing systems. The final section of this chapter explains the different ways that user expertise can be measured.

2.1 Recommender systems

Even though Recommender Systems and Question Routing techniques can be similar, not all types of Recommender Systems can be used for Question Routing. This section will discuss which types are and which aren't fit for use in Question Routing.

Recommender Systems are systems that recommend an item to a user that interacts with the system. There are two main approaches to achieve the goal of recommendation. Collaborative Filtering (CF) is an approach where the system is agnostic of all data except user preference. Content-Based Filtering (CBF) inspects the content of the items and matches this with the obtained preferences of a user.

2.1.1 Preference Expression

Both Content-Based Filtering and Collaborative Filtering depend on how users express their preference for an item. Research has shown that different types of expressed preference exists in Recommender Systems [17]. For example clicking on something is an implicitly expressed preference, while liking something is an explicitly expressed preference. In explicit expression the users consciously expresses their preferences.

Type	Description	Implicit	Explicit
Unary	Only preference or non-preference is expressed	click, buy, answer	favorite, like
Binary	Both preference and non-preference are expressed		like / ignore, thumbs up / down
Discrete	Preference is expressed as a value in a discrete range		Star rating, grade
Continues	Preferences are expressed on a continue scale		Slider green/red, slider dislike/like

Table 2.1: Different types of preference expression in Recommender Systems

In implicit expression user preferences are derived from user actions. The two main categories are listed in Table 2.1.

2.1.2 Content Based Filtering

In Content Based Filtering (CBF) items are modeled by their properties $p_1 \dots p_n$ [17] and users are modeled by their preferences. Item recommendation is done by finding the items that have properties which matches with the users' preferences. The best matches are returned as recommendations for the user. An overview on how this process works is given in Figure 2.1.

In a CBF Recommender System, finding matches for a user u can be done in a multitude of ways. Whichever way is chosen, comparing a user with an item is a required step. The items that are best comparable to the user u are provided as recommendations. In most cases the user u and item i are compared based on their similarity [17]. In the case both users and items are represented by the same Vector Space Model (VSM) m , the similarity between u and i is often based on the cosine

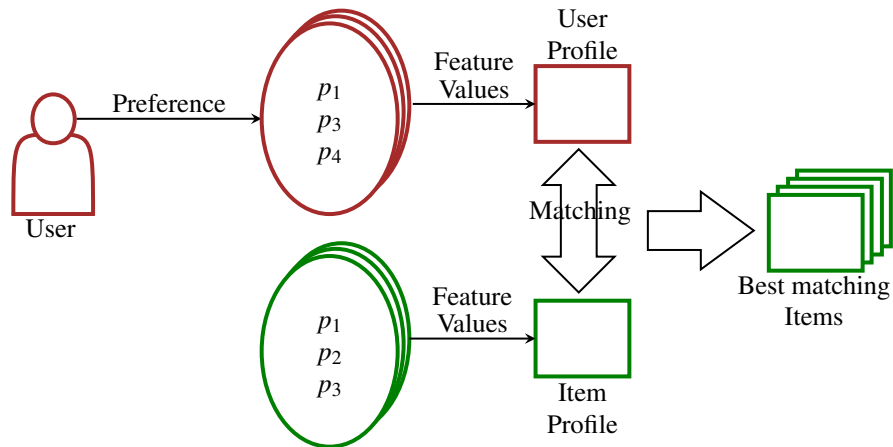


Figure 2.1: **Workflow of CBF Recommendation** In this workflow, the user profile is created based on expressed preferences on certain features. The item profile is created based on analyzing its features. Matching user feature preferences and items features results in a set of items that are recommended.

similarity which represents the closeness of two vectors in space. If two vectors are aligned in their vector space, the value of the cosine similarity is higher than when they are not aligned.

$$\text{cosine_similarity}(u, i) = \frac{\sum_k^{[m]} m_{ki} \cdot m_{ku}}{\sqrt{\sum_k^{[m]} m_{ki}^2} \cdot \sqrt{\sum_k^{[m]} m_{ku}^2}} \quad (2.1)$$

2.1.3 Collaborative Filtering

Collaborative Filtering (CF) recommends items based on expressed user preferences [21]. The user preference is expressed either implicitly or explicitly. The expressed preference is used to guess the preference for a for the user unseen item.

Two strategies exist that utilize expressed user preferences to generate recommendations. The first is a user-based approach, where users are compared to each-other to generate a recommendation set for a given user. The other approach is the item-based approach where items are compared to each-other to generate a recommendation set for a given user. Both approaches are explained in this section.

User-based

In user-based Collaborative Filtering similarity is calculated between users based on the similarity of their preferences. If user u_1 and user u_2 both have rated items $i_{a,b,c}$ the same, but user u_2 has rated i_d as well, i_d may be interesting for u_1 as well.

In a user-based CF recommender system the set I of all items that are unknown for a given user u are considered. For each item i in I every user u_i in the set of users U_i that have expressed their preference for i is compared for similarity with user u . The predicted preference of user u for i is the average of the ratings given by the other users, weighted by the similarity of u and u_i . The items with the highest predicted preference for user u are given as suggestion.

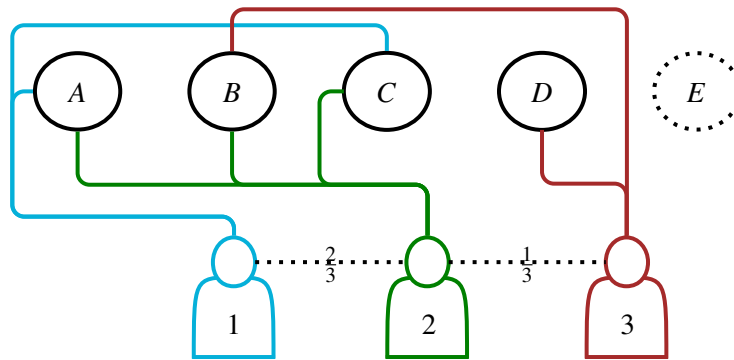


Figure 2.2: Example of user based Collaborative Filtering In this example unary preferences are given, which are represented by solid lines between the figures and the circles. The similarity is calculated with the Tanimoto similarity. In this case the new item E is not considered in the recommendation, as no user has had interaction with it. The only potential candidate for recommendation for user 1 is A with a value of $\frac{2}{3}$.

In user-based CF recommender systems existing preferences for an item of other users are required in order to predict the preferences of a user for this item. This requirement is inherent for this type of recommendation, no similarity can be calculated for a user that has no expressed preference and no preference can be predicted for an item that has no expressed preferences yet. In Figure 2.2 item E can not be recommended, because there are no known preferences. The inability to recommend new users/items is known as the ‘Cold Start Problem’ [22]. Using this approach for Question Routing poses a serious problem, because QR specifically routes new and unrated questions.

Item-based

In item-based Collaborative Filtering similarity is calculated between items. The item similarity uses expressed user preference to calculate a similarity value [21]. Users that have interacted with both item i and j are used in some similarity calculation $S(i, j)$ between item i and j . Which similarity calculation is used differs per type of expressed preferences of a user. For example when preferences are unary, the similarity between two items can be calculated by the Tanimoto[9] similarity as given in Equation 2.2.

$$S(i, j) = \frac{|U_i \cap U_j|}{|U_i \cup U_j|} \quad (2.2)$$

U_i and U_j are the sets of users that have respectively expressed their preference for item i and item j . Thus the more users have expressed their preference on both items, the higher the similarity is between the items.

Item-based CF suffers from the cold start problem as well, new items cannot be recommended because no users have expressed a preference for this item. An example of the cold start problem and how item-based CF works is depicted in Figure 2.3. In this figure item E has no expressed preference from the users and is considered new.

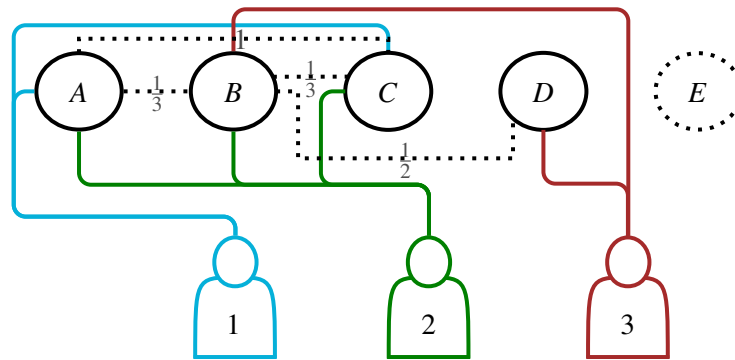


Figure 2.3: **Example of item based Collaborative Filtering** In this example unary preferences are given, which are represented by solid lines between the figures and the circles. The similarity is calculated with the Tanimoto function. In this case the new item E is not considered for recommendation, because no user has expressed preference for it. Potential candidates for recommendation for user 3 are A and C , both with a value of $\frac{1}{3}$.

The cold start problems also exists for new users. With no preference for an item, no items can be compared, thus no items can be suggested to a user.

2.1.4 Advantages and drawbacks

There are several advantages and drawbacks in Content Based Filtering recommender systems when compared to Collaborative Filtering recommender systems [17].

- + **DOMAIN KNOWLEDGE** In order to create a CBF Recommender System, it is required to analyze the domain in which the systems is expected to operate. This allows for an accurate model of both the users and items.
- + **USER KNOWLEDGE** Each user has its own preferences which are independent of other users. This allows for tailor-made recommendations based on the user profile.
- + **NEW ITEMS** Because of the item model and domain knowledge, a CBF recommender system can analyze any new item that is added to the system. The item profile can be constructed directly from the item properties which allows the item to instantly be a candidate for recommendation.
- + **TRANSPARENCY** Due to the domain analysis, user model and item model it is known on which features an items is selected for recommendation. Showing on which preferences an item is selected to the user helps to build user trust for the recommender system.
- **INFLEXABILITY** Although domain knowledge allows for an accurate user and item model, it also requires the models not to change. Any change in the item or user properties, has a high probability that there needs to be a change in the models to cope with the change.
- **NEW USERS** New users cannot get recommendations when they enter the system. Often users are even required to express preference for a few items before a reliable user model can be constructed.
- **OVER SPECIALIZATION** When the CBF recommender system only recommends items that the user has liked before, but no novel items are recommended, the system has over specialized for the user. The user ends up in a recommendation bubble of items that are not diverse.

2.2 Question Routing in CQA

Question Routing (QR) has its roots in Content-Based Recommender Systems. The Collaborative Filtering Recommender Systems are not applicable for QR, because Collaborative Filtering approaches are unable to calculate the user preference for an unseen item. Therefore Content-Based approaches are the only option. This is also the reason why in literature Content-Based approaches are so common in Question Routing systems [28, 15, 27].

The approach for Question Routing is the same as for Content-Based Recommender Systems: in both approaches a user is matched with an item. The main difference between CB Recommender Systems and Question Routing systems is in the goal each system tries to achieve. Question Routing systems return the best users given a question, while CB Recommender Systems would return the best questions for a given user.

This results in a different coverage for each approach. Content-Based Recommender Systems create recommendations for all users in the systems, thus covering all users. However it is possible that a question is not recommended to any user, thus not all questions are covered. In contrast Question Routing systems route all questions to users, thus covering all questions. For Question Routing it is not guaranteed that all users have a question routed to them.

In an ideal setting with the optimal system, there would be no difference between Question Routing and recommending questions with a Content-Based Recommender system. All users would be covered as well as all questions.

To achieve its task a QR system uses a user and question profile based on a purposely designed user and question model. The next section explains which kind of user and question models are found in QR systems and how they are constructed.

2.3 Modeling Users and Questions for Question Routing in CQA

Both users and questions are modeled in Question Routing systems. These models often show some overlap, it is easier to calculate a preference score when the models are similar. This section therefore discusses several approaches in which both the user and the question are modeled by the same approach. The difference between users and questions is in how the profile is constructed.

2.3.1 Models used in Question Routing

In Question Routing several techniques exist for modeling users and questions. From those techniques three basic concepts for the model are used.

1. Models based on a Language Model (LM)
2. Models based on a Topic Model (TM)
3. Models based on Keyword Vector Space Model (KVSM)

Language Models

The LM is a basic representation for textual content. The text is represented by its words and the number of times they occur, but the order of the words is not necessarily used. Often techniques like stop-word removal and stemming are used to reduce the number of distinct words.

Language Models originate from Information Retrieval, where they are used for retrieving a document based on a query. For each document it is calculated how likely

it is that it will ‘generate’ the query. For Question Routing the probability is calculated that a user prefers a question.

There are several different probability models that are used to estimate the probability. The simplest approach is known as the Query Likelihood Language Model (QLLM). Using Bayes rule, the probability that a user u has a preference for a question q is calculated as $P(u|q) = \frac{P(q|u)P(u)}{P(q)}$. This formula is often simplified because $P(q)$ is the same for all users and there is an assumption that $P(u)$ is normally distributed thus $P(u)$ can be disregarded as well. This results in the formula $P(u|q) = P(q|u)$.

The common way to estimate $P(q|u)$ is to use a multinomial unigram language model.

$$P(q|M_u) = \prod_{\omega \in q} P(\omega|M_u)^{tf_{\omega,u}}$$

where u is the user, q is the question, $tf_{\omega,u}$ the frequency of term ω in u . M_u is the Language model of user u . For each term ω the Maximum Likelihood Estimation (MLE) is used in the Language model. The MLE is the total number of times term ω occurs for user u divided by the total number of words that occur for user u .

This approach has some issues, one of which is that a question can have words that are not seen by a given user. This has an effect that the user gets a zero probability to prefer the question, even if other words do match. In order to combat this, several smoothing strategies have been developed. The most simple approach is to use ‘add one’ or ‘laplace’ smoothing, where for each word there is at least one observation. By adding one observation for each word, the probability will of a word will never be zero. In [15] Jelinek-Mercer[10] smoothing is applied, but other methods like Good-Turing [5], Kneser-Ney [13] and Witten-Bell [23] exist as well. More background information on smoothing for Language Models can be found in [3].

Topic Models

Topic models are models where a feature matrix is reduced to a topic matrix of smaller dimensions. In Language Models all words are used in a feature matrix to accurately approximate the underlying Language Model, this may result in sparse matrices of several thousand dimensions. One often used topic modeling approach seen in Question Routing is Latent Dirichlet Allocation (LDA) [11, 16, 6, 2].

For a topic model in general, the probability $P(t|q)$ that a question q belongs to a topic t is calculated. The user model is constructed using the information topical information. Users that have expressed preference for a question implicitly express preference for the topics of the question. In [2] the probabilities per topic of all answered questions is summed to gain a profile. Another approach is used in [16], where the number of answers per category is counted.

Keyword Vector Space Models

The model that is used for this thesis is based on the Keyword Vector Space Model (KVSM). The idea to use this model is based on regular recommendation systems, where it is used quite frequently [17]. For this model keywords or terms are extracted from the question and used in a vector representation. All terms together form a vocabulary V .

The KVSM uses a n -dimensional vector to represent questions and users where n is the size of V . The vector represents the document as a vector of term weights. Thus each term $t_k \in V$ is represented by a weight w , resulting in the document d being represented as $d = w_1, w_2, \dots, w_n$, where w_k is the weight for term t_k in the document.

One common way to create the weights for the terms is to use TF-IDF (Term Frequency-Inverse Document Frequency). This weighing scheme is often used in the field of Information Retrieval. Next to assigning weights, finding similarity is also needed. The cosine similarity is a very common similarity measure for any KVSM. Cosine similarity describes the closeness of two vectors, where closer vectors have a higher similarity.

The advantage of this model is that it is easy to create and rather easy to understand. This is also true for the cosine similarity, which is widely used in Information Retrieval and its implications are well understood. Another advantage of this model is that it is capable of turning any text into a user model, which means that it can be used for cross domain recommendation.

One of the disadvantages of this model is that it tries to get the semantics of the question, but it fails to accurately do so. A question on the same subject, but with a different vocabulary will not look the same in this model. Another disadvantage is that the dimensions of the vector are rather large, which results in longer computation times. Techniques like stop-word removal and stemming reduce the dimension, but these techniques do not significantly reduce the dimensions.

2.3.2 User model specifics

Some existing Question Routing systems extend their user model further than the models discussed so far. They often use specific features of the user to increase the probability that a user will actually provide an answer to the question.

One common feature is the number of answers provided by a user [11, 15]. Although this feature might increase the probability that a given user will answer a question, the number of answers of a user does not account for the quality of his answers.

Therefore [11] added the number of asked questions, percentage of best answers and the number of best answers to the user profile and trained a Support Vector Machine (SVM). A similar approach is used in this work, where the number of questions, user score, score per question, user expertise and user preference for a question are used to train a ranker with Coordinate Ascent.

In [26] a different approach is used. The user interest and user answer quality are modeled independent of each-other. User interest is modeled with a QLLM and use maximum entropy to estimate answer quality. This approach is similar to what is done in this work, where user interest is modeled as a KVSM and user expertise is modeled independently.

2.4 Evaluating Question Routing Systems

Since Recommender Systems are the basis for Question Routing systems, their evaluation methods have an overlap as well. A good overview for evaluation Recommender Systems is given in [8]. This section will discuss some of the evaluation methods

mentioned in [8] and will indicate how they are used for QR. But first this section introduces three different types of verification methods.

Offline verification In offline verification a dataset of known expressed user preferences is used. A Question Routing system is trained on (part of) the data to learn the preferences of the users. After the training phase the QR system routes questions from the dataset to users in the dataset. Each question q in the dataset has users that have expressed a preference for this particular question. This set of users for question q is called the ground-truth for q . Using the suggested users from the QR system and the ground-truth there are several metrics that can be used to indicate the performance of the QR strategy. The advantage of this approach is the speed of which verification can take place, which allows for an easy and cheap comparison of different QR systems.

User test In a user test the recommendations that arise from the QR system are evaluated by the users. A user test often focuses on the user interaction and user satisfaction rather than absolute performance of the QR strategy. User tests are conducted on a small set of users, who will use the system and will have to fill in a questionnaire. The results of the questionnaire indicate the user satisfaction while the observations during the interaction with the system indicates user interaction. Since QR systems are used to help users, a user test is a good way of verifying user satisfaction without risk. Unfortunately a user tests take a considerable amount resources compared to offline verification.

Online verification In online verification a new Question Routing system is placed into a live environment, often together with the system that is already in place. Users are either consistently given the old or the new strategy, this is known as A-B testing. The performance of the two strategies often can be compared on one or more Key Performance Indicators (KPIs) of the platform as a whole. A KPI for a Community Question and Answering system can be the number of questions answered, the time in which an answer is provided or the quality of the provided answers. Online verification requires that there is an existing platform where experiments can take place. Often such a platform is not available which makes this approach difficult to use.

For this research offline verification is used. This approach is chosen because it allows for different system configurations to be tested easily and relatively fast. For offline verification several metrics are available to calculate the performance of a recommendation. These metrics fall into two categories, Set-based and Rank-based. Set-based metrics do not incorporate the order of the recommended items, while Rank-based metrics do.

2.4.1 Set-based metrics

Set based metrics are metrics that do not incorporate the rank of the ground truth nor the rank of the selected candidates in the calculation.

Precision, Recall and F1 score Precision indicates how many of the selected candidates have actually answered the question. Thus it is an indication on how good the suggestions are.

Recall indicates how many of the actual answerers are in the selected candidates set. Thus it is an indication on how well the suggestions represent the full set of answerers.

F1 score is a combination of both Recall and Precision. F1 weights both the recall and the precision score, thus provides an indication on how well the suggestions are in general. The suggestions should both have a high precision and a high recall in order for F1 to increase.

Precision, Recall and F1 score are calculated per recommendation. Precision is calculated as

$$precision(q) = \frac{A_q \cap P_q}{P_q}$$

recall is calculated as

$$recall(q) = \frac{A_q \cap P_q}{A_q}$$

and the F1 score is calculated as

$$F1(q) = 2 \cdot \frac{precision(q) \cdot recall(q)}{precision(q) + recall(q)}$$

A_q is the ground truth of question q and P_q is the set of selected candidates.

In order to evaluate the overall performance of the QR strategy these values are averaged over all recommendations. Thus given a set of recommendations R that have a precision, recall and F1 score, the overall precision, recall and F1 score is defined as follows

$$overall\ metric = \frac{1}{|R|} \sum_{q \in R} metric(q)$$

2.4.2 Rank-based metrics

Rank based metrics are metrics that incorporate the rank of the ground truth and the rank of the selected candidates in the calculation.

Normalized Discounted Cumulative Gain (NDCG) NDCG is a measure that indicates how well the order of the selected candidates is compared to the order of the ground truth. If the order of the suggested candidates is exactly the reverse of that of the order of the ground truth, the score will be much lower than if the order would be exactly the same.

If the selected candidates are only a subset of the full ground truth, the NDCG metric can still give the optimal score. For instance if the ground truth is $\langle 1, 2, 3, 4, 5 \rangle$ and the selected candidates are $\langle 1, 2, 3 \rangle$, the NDCG is optimal. If the selected candidates are $\langle 2, 1, 3 \rangle$ the NDCG is non-optimal. NDCG can thus be seen as a specialized version of precision, although their scores may be very different.

The formula for NDCG is based on the Discounted Cumulative Gain (DCG) and the Ideal Discounted Cumulative Gain (IDCG)

$$NDCG_n = \frac{DCG_n}{IDCG_n}$$

There are multiple definitions available in literature for DCG. The main differences between these definitions is the weight of the discount for an incorrect order. Some definitions allow the first and second place to be swapped without a consequence in the resulting score. For Question Routing this is not satisfactory, as in general there may be several good answers but only one is the best. The definition of DCG in this work therefore is a strict one and was first suggested in [1].

$$DCG_n(q) = \sum_{i=1}^n \frac{2^{rel_i} - 1}{\log_2 i + 1}$$

Since relevance rel_i is defined in the original definition, it is defined as

$$rel_i = (|A_q| - pos_i) + 1$$

For the NDCG measure it holds that if a suggested answerer is not in the ground-truth, thus $P_q - A_q \neq \emptyset$ the value for this item is 0. Since NDCG is calculated per question, the NDCG for the Question Routing strategy is calculated as the mean of the NDCG of all questions. Thus for a set of questions Q that are processed the overall NDCG is calculated as

$$NDCG_n^{overall} = \frac{1}{|Q|} \sum_{q \in Q} NDCG(q)$$

Mean Average Precision (MAP) MAP gives an indication on how well the top n recommendations were. A suggested set of candidates is sorted on their score, after which for each subset of increasing size the precision is calculated. Thus if the first is correct, the precision is 1, while if the first is incorrect the precision is 0. This approach is called the average precision

$$AvgP = \frac{\sum_{k=1}^n P(k) \cdot rel(k)}{ground-truth}$$

where $P(k)$ indicates if user k is in the ground truth, $P(k)$ is the precision of the recommendations up to k and n is the total number of recommendations. The MAP is the mean over all recommendations Q .

$$MAP = \frac{\sum_{q=1}^Q AvgP(q)}{Q}$$

Mean Reciprocal Rank (MRR) MRR is based on the actual rank of the best suggestion. It shows how good the best suggestion was. For example a user that is suggested as the best, but actually was third, has an MRR of $\frac{1}{3}$

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|A|} \frac{1}{rank_i}$$

2.5 Pitfalls of performance metrics

Although the same metrics can be used for both Question Routing and Recommender Systems, the values of the metrics can differ significantly. This is a known issue [8].

The problem is in the size of the ground truth. The size of the ground truth for Question Routing is the number of answers, which can be anything from 1 to 10. For Recommender Systems the ground truth is the list of items with previously expressed preference, which can be anything from 1 to a few thousand items. Due to the larger ground truth it is easier for Recommender systems to achieve a higher precision — the probability that a selected candidate is in the ground truth is higher. For Question Routing the contrary is true, it has a smaller ground truth which reduces the probability that the selected candidate is in the ground truth.

To illustrate this problem a random recommendation test has been done. The test randomly selects 4 candidates out of a set of 1000 candidates as the recommendations. Two different scenarios are tested.

In scenario 1 the 1000 candidates are movies; 50 movies have been selected by a user and those movies are used as the ground truth. Four different movies are randomly selected from the set of 1000 and recommended to the user. In this scenario the size of the ground truth is 50, the size of the recommendations is 4 and the size of the pool of candidates is 1000.

In scenario 2 a question has received three answers from different users which are used as the ground truth. The 1000 candidates are users that can answer a question. The question is routed to four different users selected at random from the set of 1000 users. In this scenario the size of the ground truth is 3, the size of the recommendations is 4 and the size of the pool of candidates is 1000.

These scenarios are executed 10.000 times keeping a fixed ground truth, a varying recommendation selection and a fixed set of candidates. The results are given in Table 2.2. The results show that the precision is 0.003 for a ground truth of size three, while the precision of a ground truth of size 50 is 0.049. Except from MRR, other measures do not have this effect.

NDCG and MAP both depend on the order of the recommended items. For both these values it holds that getting the first item correct gives a higher score than getting the lower ranked items right. They are in fact more dependent on the number of recommendations rather than the size of the ground truth. This is much less of an issue since the number of recommendations can be controlled, which is impossible for the size of the ground truth.

2.6 User expertise

User expertise, or expertise, has many definitions and usages in literature. Even for the literature restricted to Community Question and Answering platforms there is no single definition of expertise. A comprehensive list of different interpretations of expertise given in [18].

For brevity only two of the most used definitions will be explained here, together with a definition that is independent of the level of activity of a user.

Scenario	Size of ground truth	Precision	Recall	NDCG	MRR	MAP
1	3	0.0031	0.0041	0.0049	0.0070	0.0044
2	50	0.0491	0.0039	0.0054	0.0179	0.0040

Table 2.2: **Various measures for random recommendation @4** The values are calculated over a 10.000 runs, the recommended items are drawn from a list of 1000 items. The items in the ground truth are constant

2.6.1 User Score

User score is the experience score that the Collaborative Question and Answering system provides for a given user. For instance on Stack Overflow the user score is reputation, while on Yahoo! Answers the user score are points.

User score is used as an expertise measure in quite some research [12, 7]. User score is used because it is readily available and intuitively represents the expertise of the user. Users of the platform have only one way of knowing if an answerer is an expert, which is looking at the answerers score.

However there are some notable disadvantages of user score as an expertise measure.

First of all the user score is not topic dependent. A user that has gathered his score on topic t does not have to be an expert at any topic $t' \neq t$.

Second it has been shown that user score can be closely related to the number of answers a user has provided [24, 7]. It has also been suggested that if a user is highly active it does not mean that the user is an expert [24], although high active users and expert user can overlap.

2.6.2 Z_{score}

The $Z_{score} = \frac{a-q}{\sqrt{a+q}}$ is an often used metric for expertise which measures expertise according to the number of posted questions q and the number of posted answers a [25].

This measure tries to capture the combined answer and asking pattern and how different it is from a user that asks and answers questions with a probability $p = 0.5$. If a users answer more questions than asks, the Z_{score} is positive, if the number of questions and answers is equal the score is about 0 and if a user asks significantly more questions than answers the Z_{score} is negative.

From the definition it is clear that the Z_{score} has similar issues as the user score. It is a topic independent measure for expertise and it also favors users that show a higher activity.

2.6.3 Mean Expertise Contribution

Mean Expertise Contribution (MEC) is a measure that is created to specifically reduce the importance of the level of activity of the user in the expertise judgment [24]. In the process this approach also incorporates topic dependency for the level of expertise. Thus this measure solves both problems that exist in the most popular measures of expertise.

MEC uses three expertise factors and is related to a single topic t . The three expertise factors are:

1. Answering Quality
2. Question Debatableness
3. User Activeness

The debatableness $\mathcal{D}(q_i)$ of a question is the number of answers it has received. The answer quality of a user is the inverse rank of the answer of the user. The debatableness is normalized by the average debatableness $\mathcal{D}_t^{avg}$ on the topic t . The average debatableness is calculated as $\frac{1}{|Q_t|} \sum_{\forall q_j \in Q_t} |A_{q_j,t}|$

The MEC value for a given user u for a given topic t is

$$MEC_{u,t} = \frac{1}{Q_t^u} \sum_{\forall q_i \in Q_t^u} \frac{1}{rank(q_i)} \cdot \frac{\mathcal{D}(q_i)}{\mathcal{D}_t^{avg}}$$

where

Q_t is the set of all t -related questions,

A_t is the set of all t -related answers,

U_t are all the users that participate in discussion about t ,

A_t^u is the set for answers provided by a user $u \in U_t$,

Q_t^u is the set of questions answered by $u \in U_t$,

$A_{q,t}$ is the set of answers provided for the question $q \in Q_t$.

Chapter 3

Question Routing in CQA Systems

3.1 Common Data Model for CQA systems

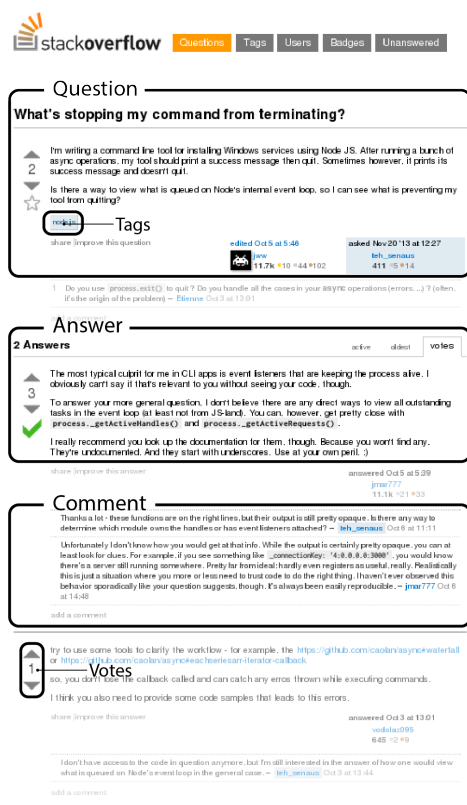


Figure 3.1: A representation of the various parts of the CQA system Stack Overflow

Even though every Community Question and Answer system is different in appearance and functionality, they do share some common entities and concepts.

All CQA platforms feature three basic entities: users, questions and answers. The users form the community and the questions and answers are the knowledge in the system. In the system users can post questions where other users can post answers to.

Next to the three basic entities, CQA often have at least three other important entities: 1) Comments; 2) Votes; 3) Tags. Comments can be placed on questions and/or answers, Votes can be given to questions and/or answers and Tags are often just given to questions.

Comments have the purpose of discussing a question and/or answer. Clarification of the question or an explanation of an answer can be requested in a comment.

Votes are used to create more user engagement, because votes help to increase reputation. Votes are also the community rating system, which allows perfect answers to bubble up, while just good answers get a lower position.

Tags are used to organize questions and help users to find relevant questions.

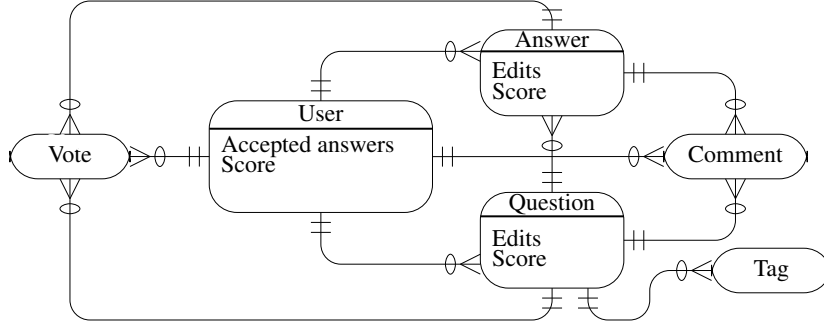


Figure 3.2: Simplified Data Model for CQA Systems.

q	Question
q_1	How to program power in Java
q_2	How to program square in C#
q_3	How to calculate power in C

Table 3.1: Example questions that could have been asked at a programming CQA

Figure 3.1 shows how comments, questions, votes, answers and users are presented on a question page of Stack Overflow. One answer is accepted, indicated by the checkmark ✓. This answer has three upvotes, two comments and only one tag. How the different features and functions interact is depicted in Figure 3.2.

3.1.1 Question Data Model

To be able to analyze questions and for the creation of a question model a number of properties p exhibited by a question are given. The provided properties do not entail the entire set of properties exhibited by a question, but mere the properties that are commonly used in the routing and analysis of Community Question and Answering systems.

In this thesis a well known feature from the field of Information Retrieval called Term Frequency — Inverse Document Frequency (TF-IDF) [20] is used.

The Term Frequency (TF) is the number of times a specific term t occurs in a document. A common way to normalize the TF is to scale the number of occurrences of a term by the total number of terms in the document.

For example in question q_1 from Table 3.1, ‘How to program power in Java’, the term *program* occurs once, thus the TF is equal to 1. Since the total number of terms in q_1 is 6, the normalized version has a TF of $\frac{1}{6}$, which is shown in Table 3.2.

The Inverse Document Frequency of a term t is defined as $IDF(t) = \log \frac{N}{DF}$ where N is the total number of documents in the corpus and DF is the Document Frequency of term t , the total number of documents that contain t at least once. In Table 3.2 all non-zero values of DF , TF and $TF - IDF$ are given for the documents in Table 3.1.

Properties of questions

p	word	DF	TF_{q_1}	TF_{q_2}	TF_{q_3}	$TF - IDF_{q_1}$	$TF - IDF_{q_2}$	$TF - IDF_{q_3}$
p_1	how	3	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$			
p_2	to	3	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$			
p_3	program	2	$\frac{1}{6}$	$\frac{1}{6}$		$\log \frac{3}{2} \cdot \frac{1}{6}$	$\log \frac{3}{2} \cdot \frac{1}{6}$	
p_4	power	2	$\frac{1}{6}$		$\frac{1}{6}$	$\log \frac{3}{2} \cdot \frac{1}{6}$		$\log \frac{3}{2} \cdot \frac{1}{6}$
p_5	in	3	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$			
p_6	Java	1	$\frac{1}{6}$			$\log 3 \cdot \frac{1}{6}$		
p_7	square	1		$\frac{1}{6}$			$\log 3 \cdot \frac{1}{6}$	
p_8	C#	1		$\frac{1}{6}$			$\log 3 \cdot \frac{1}{6}$	
p_9	calculate	1			$\frac{1}{6}$			$\log 3 \cdot \frac{1}{6}$
p_{10}	C	1			$\frac{1}{6}$			$\log 3 \cdot \frac{1}{6}$

Table 3.2: The TF and DF of every term and every document from Table 3.1.

TF-IDF all words Questions on a CQA platform often have a title and body. The title often is the question itself, while the body provides the extra information and context for the question. These pieces of text combined can be converted to a TF-IDF vector. This vector then describes the unique words of this questions, which helps to differentiate this question from other questions. Even though it describes the uniqueness of a question, this feature may contain noise. This can happen when there are many unique words and thus each question is extremely different from other questions according to cosine similarity while in fact they may be semantically very similar.

TF-IDF of tags In CQA platforms each question is tagged with one or more tags. The user creating the question can attach up to a certain amount of tags to the question. The tags given to a question can provide the needed input for a TF-IDF calculation.

The tags are the terms and each question is represented by its tags. Compared to the number of words there are often significantly fewer terms in the tag based approach. This may be an advantage as there is a higher chance that there are some similarities in tags and thus that questions are similar. A disadvantage may be that although the tags are similar, the semantics may be significantly different for the question.

Structural properties Structural properties are properties that in no way encompass the semantics of a question, but that are mere statistics that can be gathered. These statistics can be, but are not limited to, the number of words, the number of tags and the number of answers. For a question two groups of structural properties are defined which will help in separating structural properties that can only be used for representing the dataset and structural properties that can be used to both represent the dataset as well as provide use as a question model. The two groups are *static* and *dynamic*.

STATIC consists of statistics that are available before a question is posted publicly on CQA platform. Examples of *static* properties are the number of words used or the user asking the question. All *static* statistics gathered are listed in Appendix B, Table B.1.

DYNAMIC are statistics that can only be gathered after a question has been posted and often change over time. Examples are the number of views or score received. All *dynamic* statistics gathered are listed in Appendix B, Table B.2

Each statistic can be a property p in the question model. The value for a property p is the value of the extracted statistic, although the values can be normalized if required. Thus for question q_1 from Table 3.1 the ‘body length’ is 26 while the ‘body word count’ is 6.

Modeling Question for Question Routing Now that several properties of a question are introduced, a model that can be used for QR can be described.

The model used for the QR system is a vector of the properties of the question. Such properties can be the TF-IDF of a word in a question or the number of tags used in a question. Thus given n properties, the question model $Q = \langle p_1, p_2, \dots, p_{n-1}, p_n \rangle$ where p_n is the value of the n^{th} property. The values of the properties are normalized to fall into the range $0 \leq p_n \leq 1$. Normalization is necessary to prevent that a single property will become the dominant property of a question.

Defining the question model as a vector of the question properties allows to easily combine different properties of a question into a single question model. This allows for an easy comparison of the attribution of different properties to the quality of the recommendations.

3.1.2 User Data Model

For the analysis of users some properties that users exhibit need to be defined. These properties may also form the basis for a user data model. The properties considered are all statistics and these are not divided into dynamic and static. These statistics do not include a textual user profile, since these are not available for some CQA platforms or are only sparsely given. In fact user profiles may indicate false preferences for various reasons [4].

Using only statistics on user behavior allows for a better insight into the user. The statistics provide insight into actual actions and user interest. This insight is very useful for any Question Routing approach of which the goal is to predict user behavior and user interest.

Each statistic is considered a property p for a given user u . The statistics chosen are listed in Appendix B, Table B.3.

Modeling Users for Question Routing

The user model for Question Routing is partially based on the question model as described in the previous Section 3.1.1. The user model exists of a vector $M_u = \langle p_1, \dots, p_{n-1}, p_n \rangle$ where p_n is the n^{th} property from the question model. The length of the vector is the same as the length of the question model vector.

To generate the user model, the question model is used as a basis. The way to use the questions to generate the user model can vary, but in this thesis the approach is to update the user model for each question a user u has expressed preference for. The

update function is

$$M_u = \frac{1}{|Q_u|} \sum_{q_i \in Q_u} q_i \cdot P_{u,q_i}$$

where Q_u is the set of question models user u expressed preference for and $q_i \in Q_u$. P_{u,q_i} is the preference value of a user u expressed for q_i .

The arithmetic mean is used because it gathers user interest over time, thus recurring interests will receive a greater value than sporadic interests.

This does not mean that the arithmetic mean is the best update function. It has been shown that user interest changes over time[14], which means that the arithmetic mean would keep old users interest far to long. The rate of adaptation is very slow for the arithmetic mean.

3.2 Framework for Developing QR Systems

A reference process for creating Question Routing systems has been developed in order to be able to efficiently test the effect of different user and question models.

The reference process consists of three stages. For each stage in this process a different strategy can be created. Figure 3.3 shows the three stages and how they are related. Each stage in the reference process has a specific function in Question Routing.

In this section the relation between the three stages are discussed, as well as the purpose of each stage. The following three sections discuss the details of different strategies created for each stage, but first a short explanation of each stage is given.

1: The preparation stage prepares the data for the next stage. The preparation can include anything from just getting users and question as well as text processing, graph creating, word clustering and other techniques to find structure in the data.

The preparation is not only for question data, but user data may be included as well. The details of the preparation stage are discussed in Section 3.3.

2: The matching stage creates and matches user profiles with question profiles. The input for this stage is the prepared data from the **preparation stage**.

The matching stage is responsible for creating the question profiles and the user profiles. The question and user profiles are used for matching user preferences with the topic of the question. The matching between the user preference and question topic can be calculate in a range of different ways. This is why this stage is separate from the other stages. This stage outputs a list of users and their matching score for every question evaluated. The different strategies will be explained in Section 3.4.

3: The ranking stage ranks users that have the highest probability that they can provide a good answer to the question. To rank the users their matching score of the **matching stage** is used, but other information about the user may be used as well. The simplest approach to ranking is to order the users by their matching score.

The output of this stage is an ordered and filtered list of users that have the highest probability to provide a good answer for a given question.

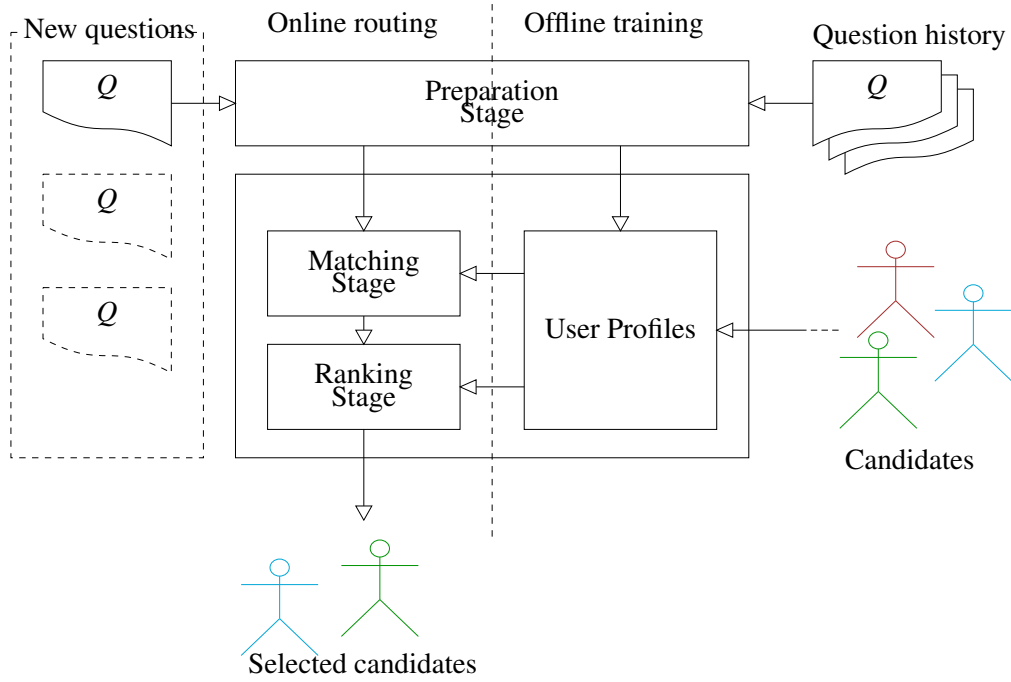


Figure 3.3: **The pipeline for creating various QR systems** Each stage in the pipeline is a location for a strategy. For each stage different strategies can be created, resulting in a flexible and diverse QR system.

The ranking stage is an addition that has not been used before. Ranking has been used in a different stage in [26], however this was not mentioned specifically.

3.3 Preparation Stage

For the data preparation stage two strategies are developed. The first strategy prepares the question text to create a keyword list. The second strategy prepares the question tags to create a keyword list.

3.3.1 Preparing question text

The goal of preparing the question text is twofold.

First of all it is used to filter noise from the question. For instance HTML does not convey any information about the content of the question, it is only used for the markup of the question. This is also true for words that do not carry (much) content related information like ‘i’, ‘we’, ‘any’ etc.

Second, when using a Keyword Vector Space Model (KVSM) the dimensions of the vector can increase dramatically when using just the unprocessed words as keywords. An example of this is shown in Table B.5 where stemming and stop-word removal is done. The initial KVSM has 11 dimensions, while after processing the KVSM has just 6 dimensions. This is a reduction of almost 50%.

The process pipeline of text preparation is depicted in Figure 3.4. As mentioned before the text preparation stage has been created to remove noise but retaining information.

The preparation steps

0. The first step is the original text.
1. The HTML is removed because it does not related to the question content.
2. The non alpha-numeric characters are removed because they do not convey information. Most non alpha-numeric are used to increase readability, which does not provide information related to the question content.
3. This is also true for capitalization of words, which is why all words are lower-cased.
4. Stopwords are word that are known to be so common in a given language that they are unlikely to carry any differentiating information, thus they can be removed without information loss.
5. Stemming is a process that reduces a word to its ‘stem’. The concept behind this is to reduce different forms of a word to a single token that represents the meaning of the word. For instance the words ‘registry’ and ‘registries’ represent the same concept, but are different words. Stemming creates for both versions of the word a single token ‘registri’, which represents the concept of a registry, but not the word registry.
6. Short terms are removed.
7. Numbers are removed as well.

To further reduce the noise and vector size, all words that occur less than n times are removed, because these words do not carry any differentiating information. This is also true for words that occur in only m questions. If a word only occurs in one or two questions, it is very unlikely that it will be occur again in a new question and thus will only increase the noise in the user profile. In this work the values for m and n respectively are 3 and 2.

3.3.2 Preprocessing question tags

Since there are not many tags, any information that is conveyed in them is likely to be valuable and thus should be used. Therefore the tags for the question are not processed at all, each tag is considered as a token.

3.4 Matching Stage

The matching stage is composed of a strategy to match users with questions. In this section the design of several types of content based Question Routing strategies are

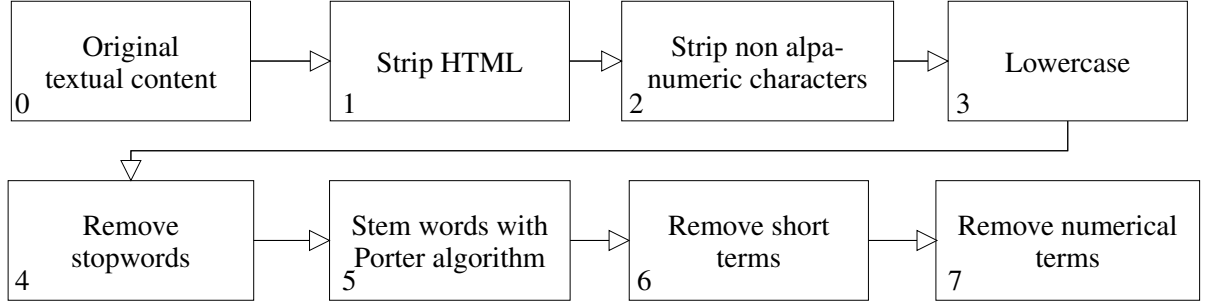


Figure 3.4: **The pipeline for processing text.** *The pipeline for processing text. A specific example on input and output of the different stages is given in Appendix B, Table B.5.*

discussed. All of the discussed strategies are evaluated for performance in the experiments chapter. A description of a matching strategy ends with a simplified formula or pseudo code of the matching algorithm. In these formula's U is the set of candidates, q is a question and $S \subseteq U$ the set of selected candidates. The general process of matching can be described as

$$S = \text{match}(q, U)$$

3.4.1 Random

The most basic Question Routing matching strategy that can be thought of is a random QR strategy. Given a question and a set of users, this system randomly picks the required amount of users from the set of given users and returns them as candidates. This QR strategy is included as a baseline. Any QR system should perform better than random.

The random QR strategy is a special case because it actually is a ranking strategy, because it randomly ranks the users. But using this as a ranking strategy would effectively remove the need for a matching stage. Therefore the Random QR strategy is placed in the matching stage.

$$\text{match}(q, U) \leftarrow \text{random}(U)$$

3.4.2 Active Answerer — AA

The Active Answer strategy counts the total number of answers each user has provided on the platform. In this strategy the matching score is the total number of answers, which is used in the ranking stage. AA is included to provide a more intelligent baseline.

This approach is not a viable real word approach because the user matching score is independent of the question and all questions will be routed to the same set of users.

$$\text{match}(q, U) \leftarrow \text{count_answers}(U)$$

3.4.3 Content based

Content based matching strategies inspect the question content to generate a match. For this several strategies exist to inspect the content of the question.

Active Interest — AI

The Active Interest strategy is more sophisticated than just counting total answers provided per user. For this strategy the number of answers per tag is calculated for each user. When the Question Routing system is given a question q , all tags T are extracted. For all users the number of answers previously provided for tag $t \in T$ are counted and the total count over all tags in T is summed. Now the user with the highest answer count are provided as candidates.

This strategy is considered a very simple content based strategy because it is required to count the tags used for the question, but no other processing is done.

This approach may have a viable real world application, because the user ranking score does depend on the question. However this approach favors the more active users, those who have more answers. Answering a lot of questions does not have to be an indication of expertise.

$$match(q, U) \leftarrow \forall u \in U \sum_{t \in T} count_answers(u, t)$$

General Interest — GI

This strategy uses the TF-IDF of words property. For each question a TF-IDF vector is created and for each user answering this question his user profile is updated using an arithmetic mean of the TF-IDF of all questions the user has answered. When a question q is given to the Question Routing system, it creates a TF-IDF vector for this question. Next the cosine similarity is calculated between all users and the TF-IDF vector of the question. The matching score returned is the cosine similarity between the user and the given question. The strategy is similar to a search engine where a query is used to retrieve documents. The question is the ‘query’ and the users are the ‘documents’. Each user is a ‘document’ consisting of all questions the user has answered.

This strategy uses the words in the corpus, which may result in a noisy feature vector. Some words in the vector may be used outside the actual domain in which they were found. I.E. a question posted with the tag C may contain ‘power’, but a question posted with the tag Haskell may contain ‘power’ as well. This would result in a Question Routing system that routes question to users that do not have the domain knowledge to answer, but have answered those type of questions before.

$$match(U, q) \leftarrow cosine_similarity(U, q)$$

Unique Interest — UI

The Unique Interest differs only in one way from the General Interest strategy. Instead of the words in the question, the attached tags are used to create a TF-IDF vector for the questions.

This strategy uses a community curated set of tags that are attached to a question. Tags may be replaced by something similar like categories or topics.

The TF-IDF processing for this approach helps to filter out topics that are applied to many questions, while boosting those topics that are used only in a small set of questions.

It is expected that this approach adequately models the user interest in certain topics. However this may result in a situation where the answerer is unlikely to provide answers for other topics which reinforces his current topics. The answerer gets trapped in a 'filter bubble'.

$$match(U, q) \leftarrow cosine_similarity(U, q)$$

Special Interest — SI

The Special Interest simply concatenates the vectors of both the General Interest and the Special Interest into a single vector. The cosine similarity is calculated on the concatenated TF-IDF vector of the GI and UI vector.

Due to the combination of the Unique Interest (UI) and the General Interest (GI) it is expected that this approach would perform the best. The answerer will not become trapped in a 'filter bubble' thanks to the broad range of GI while still being matched with the right questions thanks to the UI part of the vector.

$$match(U, q) \leftarrow cosine_similarity(U, q)$$

3.5 Ranking Stage

Ranking is the third stage in the reference process for creating QR systems. The results from the `matching` stage are taken and users are ranked again based on their matching score and potentially other properties.

In general ranking can be described as

$$R = rank(S)$$

where S is the result of the original QR matching strategy, $rank$ is a function that orders and returns the set $R \subseteq S \subseteq U$.

There are four ranking approaches used in this work.

The first strategy is to sort the users based on their matching scores of the `matching` stage. This is a typical classic approach.

The second and third strategies use a linear combination of the user matching score and user expertise. Thus

$$rank(u) \leftarrow \alpha \cdot e_u + (1 - \alpha) \cdot match(u, q) \quad (3.1)$$

where e_u is a user expertise feature of user u and $match(u, q)$ is the matching score between user u and question q . The weights of α can be varied between 0 and 1 to tune the performance of the measure. A danger of optimizing α by varying its value is that the found alpha works extremely well for one dataset, but may be very unfitting for a different dataset.

The goal of using such a ranking method is to be able to directly compare the effect of a single user feature. Since this work focuses on expertise, different measures of

expertise can be used to calculate the final ranking of the users and their effect on QR can be compared.

The fourth strategy uses a machine learning algorithm for learning to rank.

User Score Ranking User Score Ranking (USR) uses the user score of the Community Question and Answer Platform as a second value next to the matching score.

The user score is normalized using linear normalization $norm(u_s) \leftarrow \frac{u_s - \max(u_s)}{\max(u_s) - \min(u_s)}$ where $\max(u_s)$ is the maximum user score in the system and $\min(u_s)$ is the minimum user score in the system. $norm(u_s)$ is set as e_u in Equation 3.1.

User score is chosen because it is the only value that is available for every user on the platform to get an indication of the expertise of a user.

MEC Ranking MEC Ranking uses the MEC value [24] as a value for ranking. MEC is chosen because it is a user activity independent measure for expertise. The MEC value is the e_u in Equation 3.1.

MEC is calculated per tag, but questions have can more than one tag. Therefore the MEC value for each tag of the question is summed to obtain the final MEC value for a user question pair. This score is then normalized using the same normalization approach as the User Score Ranking strategy.

Learning to Rank The fourth approach for ranking is a learning to rank approach. In this approach the result of *match* is used together with several properties of the user.

Learning to rank is trained to learn the weights of the properties that are given to it. The weights are learned by providing a ranked list of items and their properties. In QR it would require a list of answerers for a question and the properties that exist between the answerer and the question.

The advantage of learning to rank is that is an automated process that can be easily trained for different datasets and is capable of fine tuning multiple weights while training. One disadvantage is that it requires a proper training set, if there are to few ranked lists it may be hard to learn good weights for the properties.

3.6 Evaluation

Most of the proposed QR strategies require a training phase to build a user profile. In order to evaluate the performance, a test phase is necessary as well. Therefore a dataset containing questions and their answers which have been asked between T_s and T_e is cut in two parts at T_c . This creates two non-overlapping sets of questions and their answers. Set $\mathcal{R}$ is the set of questions between T_s and T_c , and all answers on those question including those answers that are provided after T_c . Set $\mathcal{E}$ is the set of questions that are asked between T_c and T_e , but includes any answers provided after T_e if the dataset allows for this. Figure 3.5 shows the timespan used to select the questions and the timespan used to select the answers which are used to create both set $\mathcal{R}$ and $\mathcal{E}$.

Set $\mathcal{R}$ is used for training the QR strategy, while set $\mathcal{E}$ is used for evaluating the performance of the strategy.

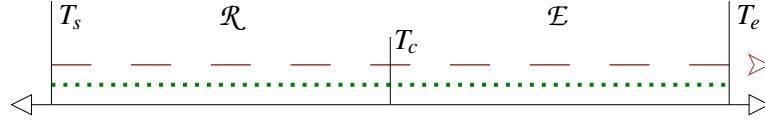


Figure 3.5: A time-line indicating which question and answers are taken as a subset. The subset selection is based on selecting all questions posted between T_s and T_e . The division of the data is done at T_c . The dotted line represents the questions considered, the dashed line represents the answers considered.

All answers are included, even if they don't fall into the time period a question was asked. The reason for this is that, especially for set $\mathcal{E}$, the focus is on finding the right users for any given question. The fact that a user took a long time to answer does not necessarily reduce the quality of the answer.

3.6.1 Data Filtering

During the testing there is a possibility that a user in the ground truth is unknown to the system. This problem is known as the cold start problem and it can have a significant impact on the performance metrics. Because the cold start problem is a different issue, all questions and answerers are filtered such that the ground truth of a question only exists of users that have answered a question in set $\mathcal{E}$. This way the cold start problem is eliminated for this dataset.

In order to be able to create a good user profile a user must have shown enough activity to learn from. For instance the user must have answered enough questions, asked enough questions or commented on enough questions. It is much more likely that a user that has shown interest in 20 question with the tag 'QR' has a preference for 'QR' than a user that has only shown interest in 1 question with the tag 'QR'. The more interest expressions the user has provided, the better the estimation of the users' preference will be.

To help in both filtering 'cold start users' as well as in filtering users without enough preference a measure called *intensity* has been developed. Details of this measure are in Section 3.6.2.

Another issue that may arise during the evaluation is that the size of the ground truth is very small. Especially after filtering, many questions will have only one answer. This will negatively impact measures like precision if the QR system always recommends $n > 1$ users. To combat the negative effect, a measure called *discussion factor* is suggested. The discussion factor filters all questions in set $\mathcal{R}$ and set $\mathcal{E}$ that have fewer than n answering users.

Intensity and *discussion factor* are not mutually exclusive. It is possible to create a combination of both sets to measure for the best possible performance. Figure 3.6 shows a Venn diagram indicating the overlap.

3.6.2 Intensity

For *intensity* there are two basic requirements:

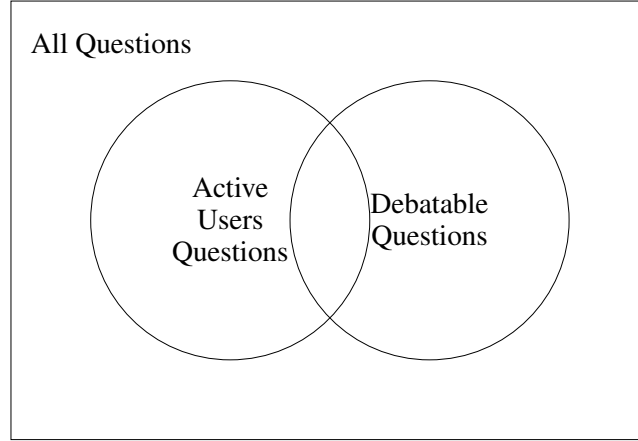


Figure 3.6: **Venn diagram with Active User Questions and Debatable Questions** *Active User Questions* is the set of questions where active users have provided answers for. The *Debatable Question* set is the set of Questions that are considered debatable.

1. The user has provided at least one answer in set $\mathcal{R}$. This way the system is able to create a user profile of the user and we can ignore the cold start problem.
2. The user has provided at least one answer in set $\mathcal{E}$. This way during verification we are sure that a perfect QR system would achieve the perfect score.

The last requirement is a tricky one because it filters candidates that have a user profile, I.E. users that have answered in set $\mathcal{R}$, but who did not answer a question in set $\mathcal{E}$ are filtered. This is an unwanted effect because if there is enough information to build a good user profile, the QR system should know if a user is interested or not. Removing these users prevents the QR system to ever misroute these users, which implicitly increasing the measured performance.

However the effect on the measured performance may be of less significance if a random subset $\mathcal{M}$ is selected from set $\mathcal{E}$ to use for evaluation. Set $\mathcal{M}$ does not have to contain all answerers from $\mathcal{E}$. Thus when evaluating on $\mathcal{M}$ there are users that have answered in set $\mathcal{R}$, but who did not answer in $\mathcal{M}$. Thus there are users with a user profile, that are not in the evaluation set. This allows the system to misroute those users

Intensity is based on two parameters, a number of time slots T_i and a number of required time slots T_r . For T_i it must hold that it is larger than the number of time slots $T_{\mathcal{E}}$ that overlap with set $\mathcal{E}$. It must also be smaller or equal to the total number of time slots T_i . Thus $T_{\mathcal{E}} < T_i \leq T_i$.

The last requirement is obvious, a user can never be required to be active in more time slots than exists. The first requirement forces the users to be active in both set $\mathcal{R}$ and $\mathcal{E}$, which makes *intensity* meet its requirements.

For example the period between T_s and T_e can be divided in to 6 equally spaced time frames. Figure 3.7 is an illustration of such a division. The actual activity requirement T_i can vary somewhere between active in at least 4 and at most 6 of those

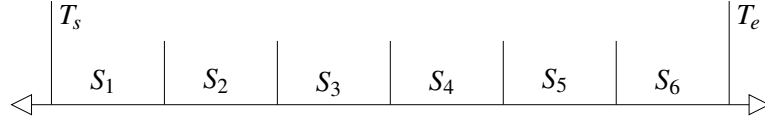


Figure 3.7: **Dividing a time period in 6 interaction slots.** The slots are of equal size, do not overlap and are between T_s and T_e .

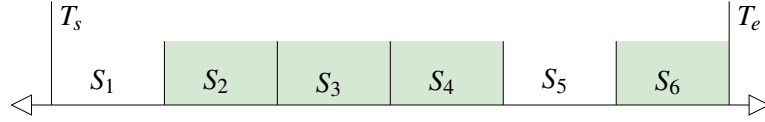


Figure 3.8: **An example of a valid 4/6 partitioning for active users.** The green slots indicate that a user has been active in the time slot that is defined. In this case $S_a = \{S_2, S_3, S_4, S_6\}$

slots. In this configuration one is guaranteed that at least one activity is shown in both set $\mathcal{R}$ and $\mathcal{E}$. An example of a user that is considered active is shown in Figure 3.8.

Formally the `intensity` function can be defined as $intensity(T_t, u, S)$ as

$$\begin{aligned} in(T_t, u, S) = & \forall Q_i \in S_a : Q_u \cap Q_i \neq \emptyset \\ & |S_a| \geq T_t \\ & S_a \subseteq S \end{aligned}$$

where

u is the user

S is a set of equally spaced time slots between T_s and T_e , each time slot contains a set of questions Q_S in a time delta $\frac{T_e - T_s}{T_t}$, where T_t defines the number of slots. Time slots do not overlap, thus the intersection of any two time slots is empty.

Q_u is the set of questions user u has provided answers for, thus showing activity.

T_t The total number of time slots available

In words, a user u is active when it has interacted with at least T_t questions that are in T_t different time slots out of T_t time slots.

3.6.3 Discussion factor

`Discussion factor` is a measure that provides an indication of the relative amount of answers for a question. The `discussion factor` is directly related to the number of answers a question has received. The definition is $df(q) = |A_q|$ where A_q is the set of answers for question q .

It is possible to transform the `discussion factor` to a normalized version where it is easier to see how discussed a question is compared to other questions. Given a

set of questions Q the normalized discussion factor is $ndf(q) = \frac{|A_q|}{avgdf(Q)}$ where $avgdf(Q) = \frac{1}{|Q|} \sum_{q \in Q} |A_q|$ is defined as the average discussion factor of the set of questions. This is useful for creating a distinction between tags. One tag may have more discussion than another tag.

For example a question about comparing two languages has two tags, C# and Java. The average number of answers of questions tagged C# is 2, while for Java it is 3. The question in the example has 5 answers. Therefore it is rather debated, this shows by having a greater than 1 for both tags, $\frac{5}{2}$ for C# and $\frac{5}{3}$ for Java.

Chapter 4

Experimental Evaluation: Question Routing on Stack Overflow

In order to measure the effect of user expertise in Question Routing, the proposed systems must be tested. In this section a description is given of Stack Overflow, the Community Question and Answer platform that the evaluation is executed on. A motivation is provided for different settings for the evaluation process in Section 4.1. Each of the settings is discussed in Section 4.2, including which dataset is used, how the data is filtered and which matching and ranking strategies are evaluated. Based on the described evaluation settings, the evaluation results are presented and analyzed in Section 4.3. The chapter ends with a discussion of the results.

4.1 Stack Overflow

Stack Overflow is a well known Community Question and Answering Platform. The focus of the platform is programming, in a very broad sense. Questions can be asked about different programming languages, on different libraries or on different subjects. Tens of thousand of topics (tags) are discussed in Stack Overflow, ranging from Javascript frontend problems to problems with a C backend. User can not only ask and answer questions, but also vote for answers and questions. The more votes an answer has, the higher it will be displayed on the page of the question.

Through these votes users can gain reputation, which represents a form of user score. The higher the reputation the more privileges a user gets. For instance voting requires a reputation of at least 10, commenting requires a reputation of at least 100. Driven by reputation, a small number of users answer to a majority of the questions, while most users only answer to a small number of questions. Due to this, some users will have a very complete user profile, while for other users much less information is available. Such difference between Stack Overflow users will pose specific challenges on question routing, as will be discussed later.

Another interesting phenomenon is that very few users both answer and ask question. The two roles seem to be quite strictly separated. A user either almost strictly ask a question or almost always strictly answers a question. Very few users do both

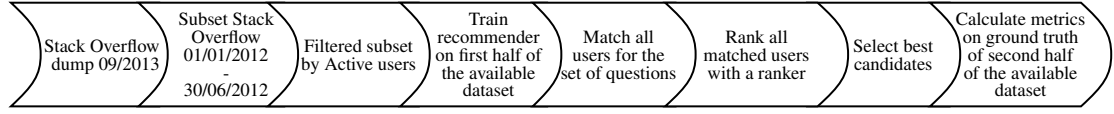


Figure 4.1: An overview of the experimental flow.

activities. Question Routing benefits the users that just want answers, while QR uses the users that just provide answers.

To evaluate the effect of question models for QR, the public dump of Stack Overflow is used. The dump contains the complete history of all questions, answers, users, votes, and badges, from the moment the website was first on-line to September 2013 (included).

4.2 Experimental Settings

Figure 4.1 shows the experimental flow. This section will explain the details of the different steps presented in the flow. Specifically, the following experimental dimensions will be motivated and described:

- The dataset selection and verification.
- The data filtering, which is done with different intensity settings
- Different matching strategies
- Different ranking strategies
- Candidate selection

First an explanation of the subset data selection is given, including a motivation that the selection is a valid representation of the entire set of data. Second the different intensity settings is discussed. For each intensity chosen an explanation is given for why this intensity is selected. For each intensity it is discussed how well the filtered data represents the actual set. The next section introduces different strategies that match a potential answerer to a question, followed by the different settings of ranking the matched answerers. In the last part it is described how candidates are selected from the ranked list.

4.2.1 Selecting a dataset

To evaluate the effect of the question models two subsets are selected from the entire dataset of Stack Overflow, based on a time span in which the question was asked. The first set, which will be referred to as set $\mathcal{A}$, spans a three month period from the first of January 2012 (T_s) and ending March 30th 2012 (T_e). The second set, which will be referred to as set $\mathcal{B}$, spans a six month period starting from the first of January 2012 (T_s) and ending June 30th 2012 (T_e).

Subset $\mathcal{A}$ is used for faster experimentation and for the configuration of the α for the ranker strategies. Set $\mathcal{B}$ is used for the evaluation, as this set is larger and represents

the full set the best. The reason for selecting a subset instead of the entire set has to do with computational issues: evaluating all data available would have required more resources than were available.

These subsets have been chosen based on time because they include a natural and sequential set of actual questions. When creating a subset, attention should be given to the representativeness of the selected data of the whole data set. If a subset is not representative of the entire set, the results may not be applicable on the entire set. To verify that the selected dataset is actually representative, several statistics have been selected to show the similarity between these subsets and the complete set *for the purpose* of Question Routing.

For each of the different stages of QR, different statistics may have an effect on the results of the evaluation. In the following paragraphs a description of influential statistics in each stage will be given. Ideally, the selected subsets should have comparable statistics discussed below with the complete set.

Representativeness in preparation stage In this stage the user profile and item profile are build. These profiles are based on different statistics that are available from the questions. There are two main contributors to the question profiles, the text of a question and the tags of a question. To ensure that the question profiles do not differ significantly these two contributors must not differ significantly. Therefore two properties for question text and tags respectively, are selected to verify that the data is similar.

The first property is the question length in characters. A longer average question length indicates that there is more information or more noise in the selected data. For a shorter question length the opposite is true. If the spread is higher, I.E. the standard deviation is higher, the information carried per question is very different. This may have an effect on the user profile which is build based on questions and thus gets its information from the questions. A higher spread may cause a higher diversity of quality information in the user profile. Similarly, the second property is for tags, the tag count and spread.

The number of answers per user is also a property that has an effect on the user profile. However this value is purposely manipulated to see how more information in the user profile changes the quality of the routing strategies.

Representativeness in Matching stage The matching stage uses the user profile and the question profile to match user interest with the topic of the question. There are no other properties than the user profile and the question profile used in the matching stage. Since those two are already covered in the preparation stage, they will not be discussed here.

Representativeness in Ranking stage The ranking strategies proposed use several different features of the user to achieve a better result.

For MEC the tags are used, as well as the average number of answers per question. The tags should have a similar distribution and the average number of answers should be the same as well in the selected subset.

	Full set	$\mathcal{A}$ — Three months	$\mathcal{B}$ — Six months
Number of users	2 332 403		
Number of questions	5 648 975	381 317	797 554
Number of answers	10 284 554	654 000	1 326 628
Mean answers per question	1.8 ± 1.6	1.7 ± 1.2	1.7 ± 1.2
Mean question length	1215 ± 1594	1225 ± 1568	1251 ± 1604
Mean tag count	3.0 ± 1.2	3.0 ± 1.2	3.0 ± 1.2
Median time to answer	19m09s	18m52s	19m08s
Median time to accepted answer	25m47s	25m49s	26m11s
Percentage unanswered questions	10.1%	6.8%	7.5%
Percentage ‘unaccepted questions’	40.0%	35.3%	36.3%

Table 4.1: **Different statistics for the used datasets.** The data is extracted from Stack Overflow.

Representativeness in Verification The dataset is also used for verification and thus the verification is also affected by which data is used.

The most important feature for the verification is the ground truth. The number of answers and their rank are used for verification. For verification it is therefore essential to have a comparable average number of answers for the questions. If the average number of answers is lower, the results from verification may be lower and if the average number of answers is higher the results may be higher as well.

Comparing the datasets Table 4.1 lists the different statistics gathered for the three datasets. The average number of answer per question is very similar, as well as the standard deviation from this value. The same is true for the tag count and the question length. From these observations it is concluded that the Question/Answer ratio between all sets are not very different. Thus the question and the user profiles are based on the same amount of information per question, even though the number of questions is clearly different.

The percentage of answered/accepted answers differs slightly, but this will not affect the results since this information is not used in the QR process.

Figure 4.2 shows the distribution of users and answers. This figures shows that few users provide many answers. For all different sets about 20% of the users provide roughly 80% to 85% of the answers. This shows that for all datasets the same skew exists.

Evaluation data Both of the selected subsets are split in two parts, a training part and a testing part. During evaluation a random sample from the testing part of the data is taken. For set $\mathcal{A}$ this is 20% while for set $\mathcal{B}$ this is 5%. This is to speed up the evaluation process. For each different setting Table B.4 in Appendix B shows the actual number of questions used for evaluation.

4.2.2 Filtering the dataset

To investigate the influence of data intensity on QR, the data is filtered according to different intensity levels. The intensity of a dataset represents how active users must

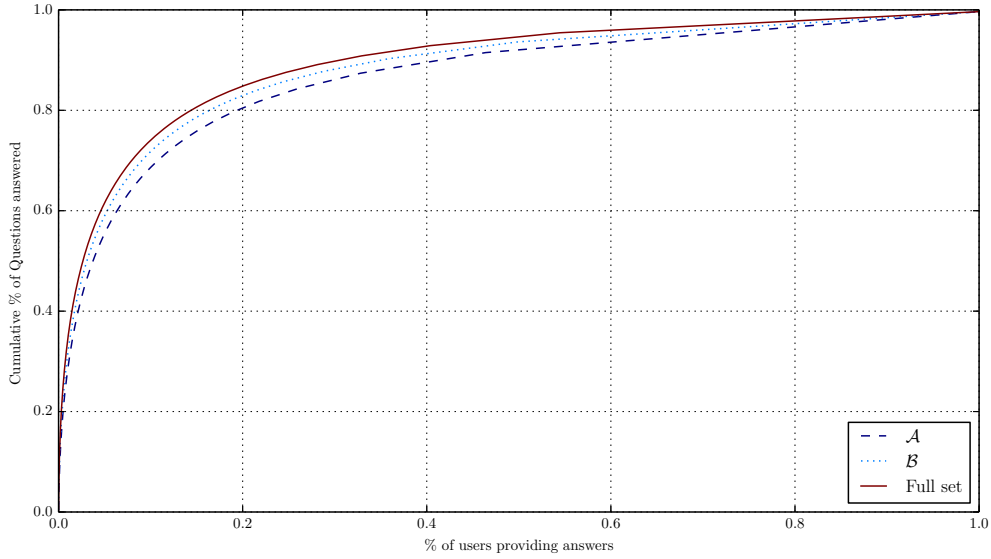


Figure 4.2: **User and answer distribution of the different datasets** This figure shows that few users provide most answers. This is similar for all selected datasets, although the distribution differs a bit.

be on a regular basis to be included in a dataset. The filtered dataset only contains questions that are answered by a user that meets the intensity requirement and only answers of those users are used.

For the experiments four different intensity levels are used. In decreasing order of strictness these levels are 12/12, 7/12, 4/6, to the most relaxed 2/2. Using the data of intensity level 12/12 it can be evaluated how the most active users change the quality of the routing result. For set $\mathcal{A}$ users have to have answered a question approximately every seven days. This results in a user having to answer at least 12 questions. Data of intensity level 2/2 approximates the actual data the best. In this set the user has to have answered a question in both the training set and the verification set. The two other sets are used to see how a decreasing intensity affects the different routing strategies.

Effect on the data The different selected users in datasets of different intensity levels lead to different questions that are used during evaluation. In order to explore how intensity affects the QR result, some statistics are presented in Table 4.2 to get an impression of how the intensity level affects the data.

Table 4.2 shows the the statistics for set $\mathcal{A}$ and $\mathcal{B}$ for the four different intensity levels. A general trend is visible that the lower the intensity is, the more the dataset represents the actual data. For set $\mathcal{A}$ with 12/12 intensity only just over 30% of the questions of the full dataset is used, while for the 2/2 intensity almost 80% of the unfiltered data is used.

Another interesting observation is that on the higher intensity levels the answers are given faster. The median time for answers for 12/12 intensity is around 12 minutes, while for the 2/2 intensity this is 17 minutes. This indicates that users that are more

Number of				Mean $\pm$ Standard deviation			Median time to	
Candidates	Questions	Answers		Answers per question	Question length	Tag count	Answer	Accepted answer
$\mathcal{A}_{12/12}$	1054	124072	353636	0.97 ± 1.62	1186 ± 1502	1.21 ± 2.95	11m50s	12m54s
$\mathcal{B}_{12/12}$	1669	262034	738493	0.94 ± 1.60	1213 ± 1534	1.21 ± 2.95	12m45s	13m50s
$\mathcal{A}_{12/7}$	6515	240635	151811	0.89 ± 1.54	1176 ± 1487	1.20 ± 2.94	14m17s	16m43s
$\mathcal{B}_{12/7}$	10481	510340	317820	0.86 ± 1.52	1203 ± 1522	1.21 ± 2.94	15m27s	18m07s
$\mathcal{A}_{6/4}$	11586	269801	415476	0.82 ± 1.47	1165 ± 1464	1.20 ± 2.93	15m16s	18m23s
$\mathcal{B}_{6/4}$	19007	568741	861768	0.79 ± 1.45	1195 ± 1508	1.21 ± 2.94	16m21s	19m51s
$\mathcal{A}_{2/2}$	25263	300004	487232	0.52 ± 1.22	1138 ± 1397	1.20 ± 2.91	16m37s	20m45s
$\mathcal{B}_{2/2}$	42967	633211	1015485	0.51 ± 1.21	1165 ± 1447	1.21 ± 2.91	17m28s	22m05s

Table 4.2: Different statistics for the filtered datasets.

active, also provide answers faster. Table 4.2 shows the median time to answer for all different filters and sets.

4.2.3 Matching strategies

All matching strategies that are introduced in the previous chapter are tested. Each matching strategy other than AA and Ra requires a ranking strategy. The ranking strategy used is the sorting ranking strategy.

The different matching strategies used are listed in Table 4.3 together with the abbreviations used for these strategies. The results from these strategies are considered as baseline for the matching+ranking strategies.

Strategy name		Strategy name	
Random	Ra	General Interest	GI
Active Answerer	AA	Unique Interest	UI
Active Interest	AI	Special Interest	SI

Table 4.3: Matching strategy names and abbreviations The two letters are the abbreviations names that are used to refer to the routing strategy.

4.2.4 Ranking strategies

The ranking strategies are not used on their own, but in combination with a matching strategy. For this, four matching strategies are chosen to be used together with the ranking strategies. The four chosen matching strategies are AI, GI, UI and SI. The reason that these strategies are chosen is because these strategies are content-based.

For each of the four matching techniques three different ranking strategies are used. These strategies are given in Table 4.4. In total this will result in twelve different Question Routing system configurations that incorporate expertise.

Strategy name		Strategy name	
Mean Expertise Contribution	MEC	User Score	US
Learning to rank	Learn		

Table 4.4: **Ranking strategy names and short description** *The letters are the short names that are used to refer to the ranking strategy.*

The different expertise measures used for ranking, like user score and MEC are normalized before being used for the ranking stage. Normalization is necessary to ensure that the values used in ranking are in the same order of magnitude, which allows for an easier combination of the values.

One ranker strategy uses the user score. For Stack Overflow the user score is the reputation of a user, used in many researches as an expertise measure. The actual value provided by Stack Overflow will be used. However this value is calculated over the entire set, not just the subset. To compensate for this the MEC value is also calculated over the entire set and not just the subset in use.

For learning to rank, an existing library has been used. This library is called RankLib¹, which has been integrated into the Lemur² project. The library has different configuration possibilities, but only one configuration is used in the experiments. The algorithm used is the default, Coordinate Ascent, the metric to optimize is NDCG@5, normalization is done with the linear normalizer for both training and evaluating. No other settings were different from the default settings.

For learning to rank, five different features of the user are used to train and later to rank. The features used are MEC, Reputation, Total number of answers, Average Reputation per Answer and finally the Matching Score from the matching phase.

The RankLib library is trained on a ranked list per question from the training part of the set. For each question in the training set, all answerers from the ground truth are ordered based on the score received on their best answer. This ranking is then provided to RankLib as an ordered list to learn from.

α selection For the MEC and US strategies a good α values must be chosen. To get an indication on which α should be used, a different range of α is used to get an indication of which value for α produces good results.

To get the different α values set $\mathcal{A}$ is used for training and evaluation. Two different matching strategies have been tested, UI and GI. The α was changed from 0 to 1 in 0.1 increments. The matching strategy is trained on the training data. After training, α is set to a specific value. After setting α the evaluation takes place using the set value for α . The results shown in Figure 4.3 indicate the performance of different α values on the testing data. Using Figure 4.3 as a reference two values for α are chosen. $\alpha = 0.6$ is chosen for the MEC ranking strategy and $\alpha = 0.7$ is chosen for the US strategy.

¹<http://sourceforge.net/p/lemur/wiki/RankLib/>

²<http://sourceforge.net/projects/lemur/>

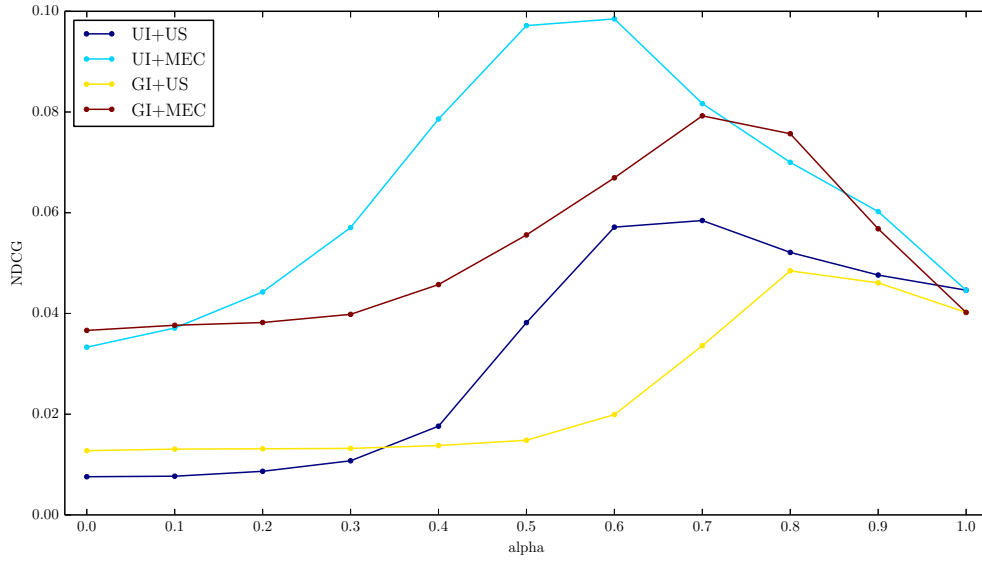


Figure 4.3: Performance for different α 's The score shown is the NDCG.

4.2.5 Selecting candidates

To be able to route the question, a selection of candidates has to be made. To achieve this for each result produced by a QR strategy the QR system picks the top 5 best ranked candidates.

Five is used because it is a realistic number to use. Most questions have 5 answers or less, thus 5 candidate answerers should be enough to find the right user that is able to answer the question. Another value could have been chosen as well, like the average number of answers per question as shown in Table 4.2. Using the average number of answers would most likely result in a better precision. Recommending 5 users when there are only 2 actual users allows a maximum precision of $\frac{2}{5}$, while recommending 2 users would allow for a maximum precision of $\frac{2}{2}$. Thus using the average number of answers would increase the precision, but in practice betting on only two to three users may be on the risky side. It is not guaranteed that users that are used in the routing are actually active.

4.3 Experimental Results

The experiments begin with setting a baseline in Section 4.3.1. To achieve this all different matching strategies are evaluated for the performance. The performance is provided as NDCG@5, but other measures have been taken as well. For those values see Table 4.5.

The different matching strategies are evaluated on set $\mathcal{B}$ and for a 12/12 intensity. In Section 4.3.2 the intensity is varied over four different levels.

The ranking is applied in Section 4.3.3 for all different intensities and for all content-based matching strategies. Section 4.3.3 therefore shows how the different

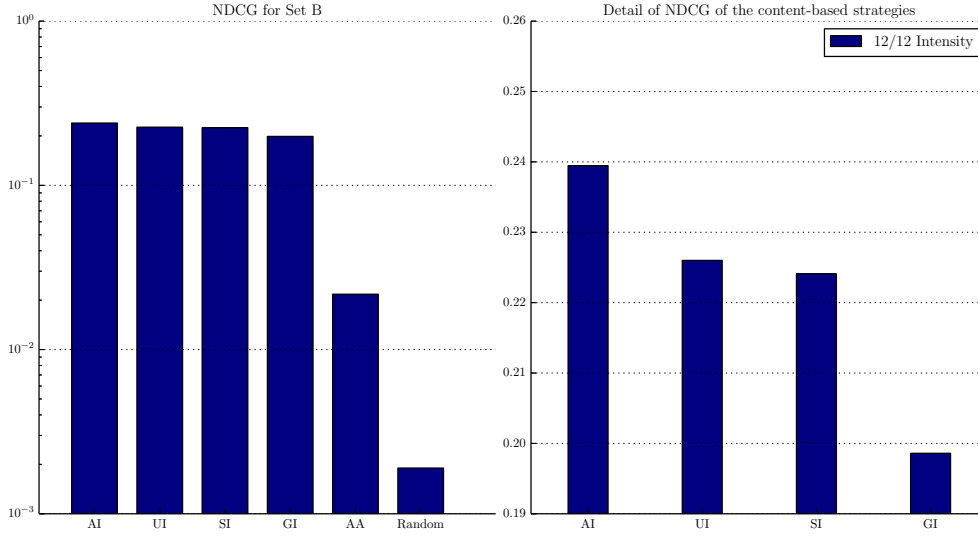


Figure 4.4: **Results for NDCG for a 12/12 intensity for a six month period, set $\mathcal{B}$.** All content-based strategies perform an order of magnitude better than the AA approach and which in turn performs an order of magnitude better than the Random approach. The performance of content-based strategies are very close in this setting.

expertise measures affect the performance of the baseline strategies.

In the last section the results are discussed. The discussion section includes a critical review on the results and an analysis on what may invalidate the results.

4.3.1 Baseline strategies

The baseline strategies used exist of strategies which do not incorporate user expertise. The performance of these strategies is used as a baseline performance indicator.

The first approach is to create a baseline for the 12/12 intensity. This baseline will show which strategies are viable and which are not. The 12/12 intensity is chosen because it is expected that all strategies will perform best on the 12/12 intensity set. The idea behind this is that there is more information per users as well as fewer users in general. This creates a greater opportunity for getting the right user to the right question.

The set used for evaluation is set $\mathcal{B}$, which is the largest set. This set has more participating users and more questions and would therefore be a more realistic example than set $\mathcal{A}$.

Figure 4.4 shows the performance of the baseline approaches. From the figure it can be read that AI performs best. This is rather unexpected as this is one of the most simple approaches. However for a high intensity it is more likely that looking at the number of answers given might be a good indication of who has answered a question.

The Ra strategy performs the worst, the results are two orders of magnitude worse than the rest. This is an indication that the extra information used in the content-based strategies increases their performance. Second-to-last is the AA strategy. This strategy

performs an order of magnitude better than the *Ra* strategy, but is still an order of magnitude worse than the content-based strategies.

However the results of *AA* are interesting, as this strategy performs much better than random, but only recommends the top 5 most answering unique users for every question. This results in only 5 users for all questions in the test set, while the *Ra* routes questions to all users in the test set. It could be argued that *AA* actually performs worse than *Ra*, because it is very unlikely that those 5 users will have time to answer all questions routed to them.

The remaining four strategies, which are all content-based, are very close together in terms of performance. It is interesting to see that *UI* outperforms the *SI* slightly. It would be expected that the *UI* which has less knowledge on the user than the *SI*, which is based on both *UI* and *GI*, would perform the worst of the two. While the actual cause is unclear, it is possible that there is too much noise in the *GI* part of the *SI* which reduces the performance.

Another issue may be that the *SI* is more sensitive for the number of answers a user has provided. This would help get it a higher performance in the higher intensity datasets.

To test if the performance of *SI* falls as well as that of *AI* the strategy is evaluated on datasets of different intensities. The other strategies are evaluated as well to see if their relative performance increase. The *AA* and *Ra* are not considered, as those strategies are not viable options.

4.3.2 Performance change over Intensity

Changing the intensity from a high to a lower level brings the filtered set closer to the reality. Changing the intensity increases the number of users in the set but decreases the information available for those users. How the number of users, questions and other values are affected is shown in Table 4.2.

The performance on the different intensity levels of the content-based strategies can be seen in Figure 4.5. There is a significant drop from 12/12 intensity to the next level of intensity. This may be caused by the reduced amount of information available per user. With an intensity of 12/7 candidates may have answered 6 times in the testing part, but only once in the training part. This users' profile thus exists of just the information of one question.

However dropping the intensity to lower values drastically increases the number of candidates available. This may also contribute to the reduced performance seen in the figure. As the number of candidates increases, the performance of all strategies decreases. It is harder to find the right answerers from all candidates.

It is interesting that *AI* is again the best performer, moreover its decrease in performance is smaller than all other approaches. The gap between the performance of *AI* and the other three approaches grows larger.

The lower decline can be explained by the fact that *AI* always picks the most answering users. If there are more users, which is the case when the intensity decreases, the users that have answered the most questions are still the same users. If the number of questions does not drastically increase the performance is likely to only slightly reduce.

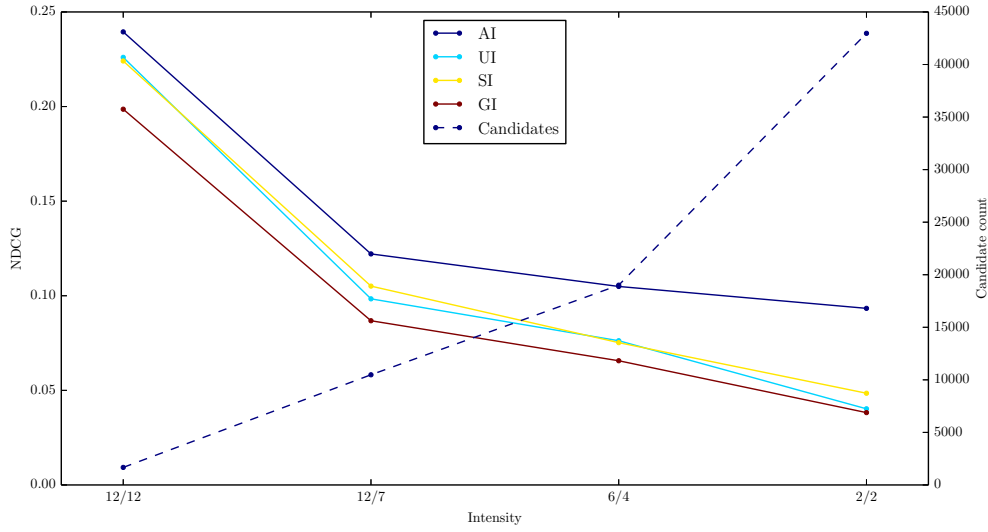


Figure 4.5: **The performance on NDCG@5 for the content-based matching strategies.** The results show the performance on different intensities for set $\mathcal{B}$. The number of candidates is shown to provide a reference on how the data changes with each intensity.

For AI this has probably happened. Initially about 900 unique candidates are chosen in the 12/12 set, for the 7/12 set about 2000 unique candidates are chosen and for the 4/6 and 2/2 have about 2100 and 2200 chosen candidates. Table 4.2 shows that the increase in questions is rather small.

In contrast, the three other content-based strategies do consider new users when available. Their information is gathered and stored. Since these strategies are less sensitive for the number of answers a user has provided their suggestions are more likely to be different if there are more candidates.

Another interesting observation is that the UI and SI perform about equally well, but when the intensity is lowest at 2/2 the SI performance is slightly better than the performance of UI.

This is what initially was expected. The GI has a broad focus and was expected to route question based on their formulation/type more than content. The UI has a very narrow topical focus and thus routes questions to users on topic, but disregarding the specific type of question. Combining both UI and GI into SI was expected to use the best from both worlds. The results show that this happens when the dataset best represents the reality. It is interesting is that GI and UI both have a similar performance on on 2/2 intensity set. However SI increases in performance and outperforms both GI and UI on which it is based.

4.3.3 Performance Increase on Expertise

The performance of the different strategies are actually very different for each intensity level. To see if expertise can stabilize the performance, different intensity levels are used for the different combinations of matching and ranking strategies.

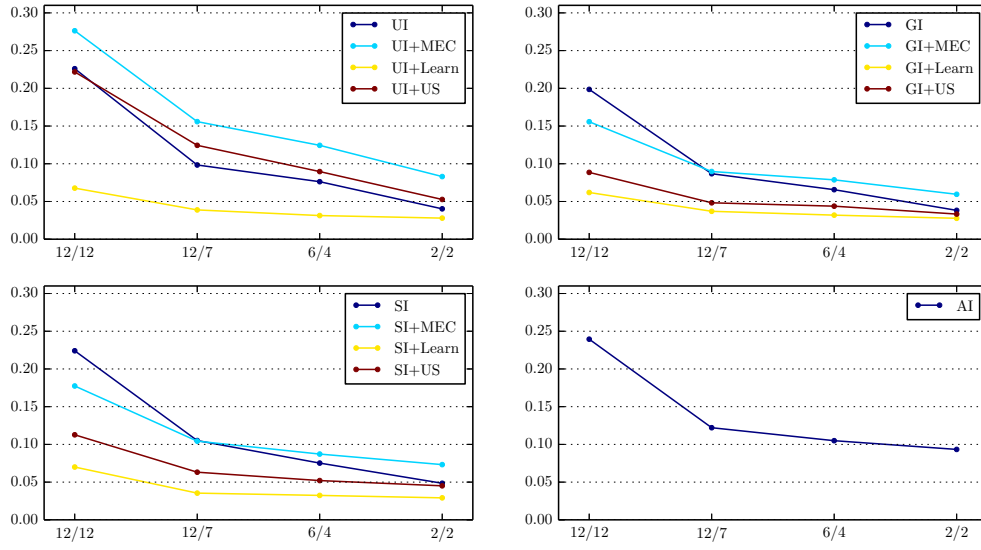


Figure 4.6: **Performance of different expertise ranking strategies.** The labels indicate the different combinations. The first part indicates the matching strategy from Table 4.3 and the part after the + sign indicates the ranking strategy from Table 4.4.

In Figure 4.6 there are four different sub-figures, one figure for each matching strategy. Each sub-figure shows the results of all ranking strategies for a single matching strategy.

From the results it can be read that adding MEC always increases the performance of the routing strategy used for the 2/2 intensity set. However the increase in performance on the entire spectrum of intensity is only true for the UI strategy. For the other approaches the MEC makes the strategy perform worse on the 12/12 intensity and perform similar on the 12/7 intensity.

The consistent increase of performance for MEC in the UI may be explained by the tuning of the α in the MEC strategy. The α is empirically set after an evaluation of different α s on a set $\mathcal{A}$ with a 12/12 intensity. The strategy used for this evaluation was UI, thus the α may be optimized for UI. It must be noted that finding the α was done on the $\mathcal{A}$ set, but the results shown here are gathered on the $\mathcal{B}$ set. Thus the data used for evaluation is not the same as the data used for finding a good α value.

The decrease in performance of the other three approaches may be explained by this same phenomena. It may be the case that their best performing α on 12/12 is very different between the different strategies. The general increase in performance for all matching strategies with MEC on the 2/2 intensity level may be due to a decreased difference in optimal α values.

MEC does reduce the rate of decline of performance for both the GI and SI strategies. It is unclear why the MEC seems to stabilize the performance. The stabilization does not happen for the UI for which the decline rate follows the decline rate of the non-ranking routing strategy closely.

Learning to rank unexpectedly performs much worse than the baseline approach. This is unexpected because this approach uses more user features and learns which

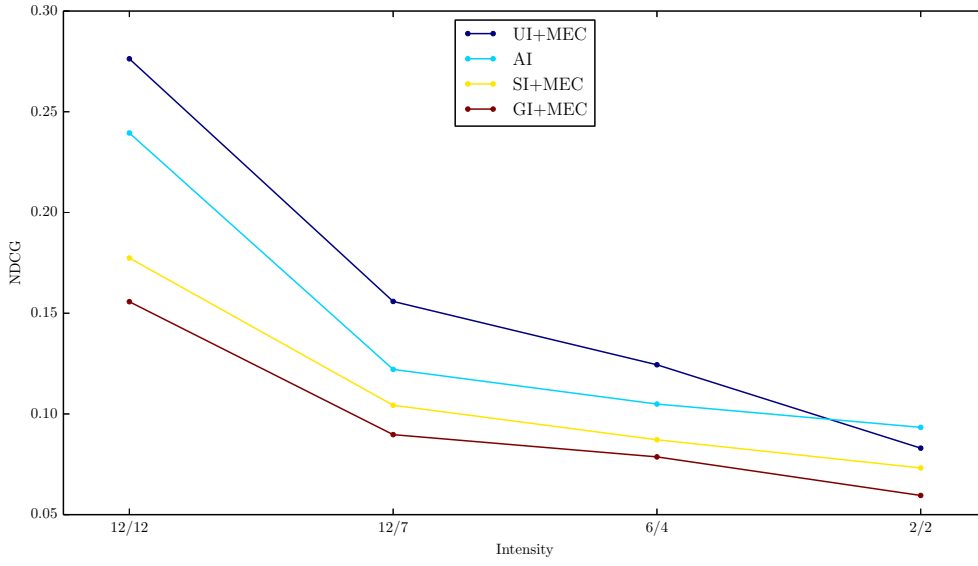


Figure 4.7: The performance of MEC compared to the best performing baseline.

features are important and which features are not.

However there may be an issue in the learning part. Even though there are many question answer pairs to learn from, a significant part of the questions have only one answer. For those there is nothing to learn, there is no comparing a ranked list. There are also quite some question with two or three answers, but those may not provide enough consistency for the algorithm to learn.

The US performs just below the baseline except for UI, which increases the performance of the baseline. It is likely that only UI benefits from US because α has been selected using UI.

The strategies that perform best on the 2/2 set are shown in Figure 4.7. This figure shows that on the 2/2 set the AI performs the best on NDCG@5. Actually from Table 4.5 it can be seen that AI not only performs best on NDCG@5, but on all other measures as well.

Figure 4.8 shows the two dimensions that are varied during testing. The different strategies are ordered based on their performance on the 12/12 intensity set. The results shown are created on the $\mathcal{B}$ set. From this figure it can be seen that reducing the intensity also reduces the performance of every routing strategy.

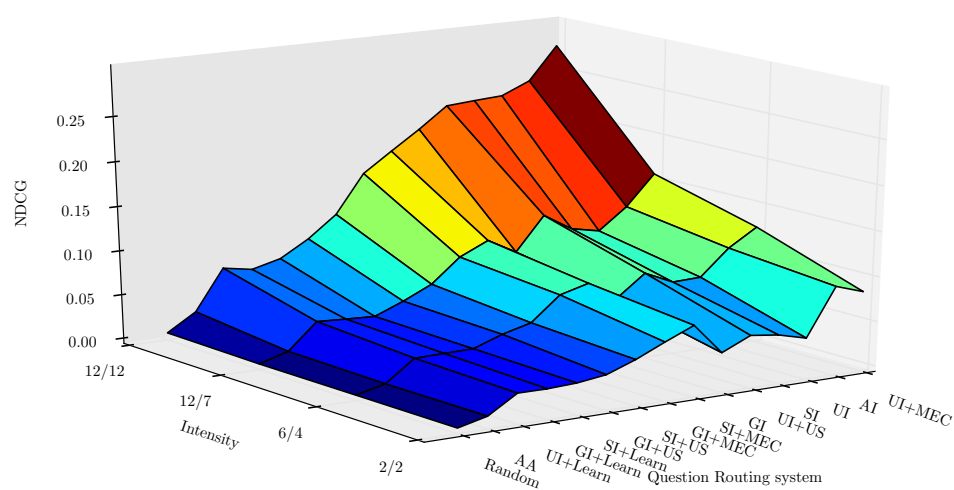


Figure 4.8: The two dimensions of testing in a single figure

Name	Ranker	P@5	%	R@5	%	F1-Score@5	%	NDCG@5	%	MAP@5	%	MRR@1	%
Ra	—	$2.52 \cdot 10^{-05}$		$4.41 \cdot 10^{-05}$		$3.06 \cdot 10^{-05}$		$8.17 \cdot 10^{-06}$		$1.21 \cdot 10^{-05}$		0	
AA	—	$2.92 \cdot 10^{-03}$		$1.09 \cdot 10^{-02}$		$4.43 \cdot 10^{-03}$		$7.83 \cdot 10^{-03}$		$5.55 \cdot 10^{-03}$		$3.98 \cdot 10^{-03}$	
AI	—	$3.34 \cdot 10^{-02}$		$1.23 \cdot 10^{-01}$		$5.06 \cdot 10^{-02}$		$9.33 \cdot 10^{-02}$		$7.64 \cdot 10^{-02}$		$5.93 \cdot 10^{-02}$	
GI	—	$1.44 \cdot 10^{-02}$		$5.53 \cdot 10^{-02}$		$2.21 \cdot 10^{-02}$		$3.82 \cdot 10^{-02}$		$3.09 \cdot 10^{-02}$		$1.97 \cdot 10^{-02}$	
UI	—	$1.49 \cdot 10^{-02}$		$6.00 \cdot 10^{-02}$		$2.33 \cdot 10^{-02}$		$4.02 \cdot 10^{-02}$		$3.24 \cdot 10^{-02}$		$1.93 \cdot 10^{-02}$	
SI	—	$1.80 \cdot 10^{-02}$		$6.84 \cdot 10^{-02}$		$2.75 \cdot 10^{-02}$		$4.85 \cdot 10^{-02}$		$3.93 \cdot 10^{-02}$		$2.68 \cdot 10^{-02}$	
GI	MEC	$2.26 \cdot 10^{-02}$	57	$7.50 \cdot 10^{-02}$	36	$3.31 \cdot 10^{-02}$	50	$5.95 \cdot 10^{-02}$	56	$3.98 \cdot 10^{-02}$	29	$3.07 \cdot 10^{-02}$	56
UI	MEC	$3.09 \cdot 10^{-02}$	107	$1.08 \cdot 10^{-01}$	80	$4.61 \cdot 10^{-02}$	98	$8.30 \cdot 10^{-02}$	106	$6.14 \cdot 10^{-02}$	90	$4.60 \cdot 10^{-02}$	138
SI	MEC	$2.73 \cdot 10^{-02}$	52	$9.18 \cdot 10^{-02}$	34	$4.02 \cdot 10^{-02}$	46	$7.32 \cdot 10^{-02}$	51	$5.11 \cdot 10^{-02}$	30	$4.07 \cdot 10^{-02}$	52
GI	Learn	$1.06 \cdot 10^{-02}$	−26	$2.79 \cdot 10^{-02}$	−50	$1.45 \cdot 10^{-02}$	−34	$2.77 \cdot 10^{-02}$	−28	$1.43 \cdot 10^{-02}$	−54	$1.63 \cdot 10^{-02}$	−17
UI	Learn	$1.10 \cdot 10^{-02}$	−26	$2.87 \cdot 10^{-02}$	−52	$1.50 \cdot 10^{-02}$	−36	$2.79 \cdot 10^{-02}$	−31	$1.45 \cdot 10^{-02}$	−55	$1.53 \cdot 10^{-02}$	−21
SI	Learn	$1.14 \cdot 10^{-02}$	−36	$3.05 \cdot 10^{-02}$	−55	$1.57 \cdot 10^{-02}$	−43	$2.91 \cdot 10^{-02}$	−40	$1.54 \cdot 10^{-02}$	−61	$1.51 \cdot 10^{-02}$	−44
GI	US	$1.26 \cdot 10^{-02}$	−13	$4.78 \cdot 10^{-02}$	−14	$1.92 \cdot 10^{-02}$	−13	$3.33 \cdot 10^{-02}$	−13	$2.58 \cdot 10^{-02}$	−17	$1.64 \cdot 10^{-02}$	−17
UI	US	$1.95 \cdot 10^{-02}$	31	$7.51 \cdot 10^{-02}$	25	$3.00 \cdot 10^{-02}$	29	$5.27 \cdot 10^{-02}$	31	$4.27 \cdot 10^{-02}$	32	$2.86 \cdot 10^{-02}$	48
SI	US	$1.63 \cdot 10^{-02}$	−9	$6.25 \cdot 10^{-02}$	−9	$2.51 \cdot 10^{-02}$	−9	$4.51 \cdot 10^{-02}$	−7	$3.57 \cdot 10^{-02}$	−9	$2.44 \cdot 10^{-02}$	−9

Table 4.5: **Experiment results for different routing strategies.** The results shown are for the $\mathcal{B}$ set and for a 2/2 intensity. The values in percentage indicate the increase in score with respect to the respective baseline approach. The bold indicate the highest score for a specific measure.

4.4 Discussion

This subsection discusses 1) some interesting observations that are obtained from experiments, and 2) the generalizability of the experimental results.

4.4.1 Insights from the Experiments

Different configurations of QR systems have been evaluated, both with and without expertise. The results of the experiments show that incorporating expertise into Question Routing can increase the performance of a Question Routing system. Of the three approaches evaluated MEC proved to increase the performance on Question Routing systems most consistently. On all different baseline approaches MEC improved the performance significantly on the 2/2 intensity set.

However other approaches were tested too, which were not as successful as MEC. User Score proved to increase performance only on a single baseline, for which it was specifically trained. Learning to rank performed the worst of all used techniques, even though it had more information available.

These results indicate that care should be taken when selecting an expertise measure. Not all measures improve the performance of the QR strategy.

Spread of routing The results have shown that MEC is a useful feature to use in QR systems. MEC consistently improved the performance of the baseline strategy. For three out of the four intensity levels evaluated, $SI+MEC$ outperformed any other tested Question Routing system. Only on the 2/2 intensity level MEC performed just below the AI baseline. However on closer inspection an interesting observation can be made. The AI baseline has routed fewer unique users to the questions when compared to $SI+MEC$. For example on the 2/2 intensity set, AI routes about 14.000 questions to just 2648 users, while $SI+MEC$ routes the same 14.000 questions to 13.430 users. So $SI+MEC$ spreads the questions on a more diverse set of users that can contribute to the knowledge in the system. Even with the increase of the number of unique users, $SI+MEC$ performs similar to AI .

Learning to Rank Learning to Rank performs far below the baseline, which is unexpected. The Learning to Rank ranker strategy was expected to perform the best of all ranker strategies thanks to the extra expertise information available.

However the training set of the `Learn` strategy contained only a small percentage of questions answer sets larger than 3. This means that most of the rankings provided to learn from, where either just 1 answer, 2 answers or 3 answers. These short lists may not contain enough information for `Learn` to learn from.

Intensity Different levels of intensity have been used for the evaluation of Question Routing systems. From the results a trend is visible, the higher the level of intensity the better the performance is for a Question Routing strategy.

The consistency of the performance of the different configurations of Question Routing systems on the different levels of intensity is not similar. Specifically MEC helps to stabilize the performance of the underlying matching strategy. For a 12/12

intensity, which is a easier routing environment MEC is useful, but in a harder environment such as with a 2/2 intensity the MEC becomes more useful and increases the overall performance of the Question Routing system.

Tags From the results it is clear that tag based approaches perform very well. Only MEC, which is a tag based measure, could increase the performance of the different matching strategies. The cause of this may be found in the fact that the easiest way as users to find a question, is by find looking at the tags of a question. The homepage of Stack Overflow for a logged in user shows ‘interesting’ questions, which have been selected “*based on your history and tag preference*”.

4.4.2 Threat to validity

Although care has been taken to do valid experiments, some decisions and assumptions are made that may provide a threat to the validity of the results. This section will list the different threats that have been identified.

Intensity The intensity filter has a two way effect. The first effect is preventing the cold start by assuring that a user has answered in the training set. The second effect is to remove users from the candidate list that have not provided an answer in the testing set. Filtering these users reduces the number of false positives. This may have an effect on the results which would be higher than if all users with a user profile where considered.

Verification An offline verification method was used. The assumption was made that users that have answered a question, are actual experts for this particular question. However this does not have to be true. This may introduce false positives in the dataset.

The other way around is probably even worse, not all experts for the question will have answered the question. Once the question has been answered, other experts will leave it be. Therefore the QR system may route a question to a user that would have answered, but did not because an answer was already given.

Dataset Although the dataset has been thoroughly analyzed to verify that it represents the actual data, it still is a rather small set compared to the full data. This may skew the results that are created on this subset.

The dataset is also a single domain from a single source. Generalizing for a different domain may not be applicable, because programming is a rather specific domain.

Chapter 5

Conclusions and Future Work

This chapter gives an overview of the project's contributions and gives suggestions for future work.

5.1 Conclusions

RQ1 *How to measure expertise?*

To answer this question a literature review was done. The literature showed that there are a number of expertise measures that exist. However most of the metrics used in literature are correlated to the number of answers a user has provided. This has been proven not to be a good indication of expertise. One measure was found that did not correlate to the number of answer given, the Mean Expertise Contribution (MEC). Other measures have been found that are used quite often are both z-score and user score.

RQ2 *How to incorporate expertise in Question Routing?*

To answer this question, a reference process was developed. This reference process exists of a three stage process that creates a Question Routing system. The first stage is data preparation, where features are extracted from the data. The second stage is matching users with questions, based on their preferences. The third stage uses the matching score and an expertise value to provide a final ranking for the users. Thus expertise was incorporated in the third stage. Setting up a Question Routing system using the reference process allows for an easier change in the used feature for expertise, while keeping the rest of the system stable.

RQ3 *How to evaluate Question Routing systems*

This question was answered by doing literature review. Since Question Routing systems have their foundation in Recommendation Systems, it was very likely that the same measures for evaluation could be used. All measures used for Recommendation System are applicable for Question Routing systems, however some pitfalls have been uncovered. The measures of precision and MRR are both rather sensitive the size of the ground truth. This has as a result that the precision of Question Routing systems

are much lower than can be expected from a Recommendation system. However other values such as NDCG and MAP are not sensitive to the size of the ground truth, thus provide similar values.

RQ4 *How does expertise influence Question Routing systems?*

An extensive evaluation of different configurations of Question Routing systems was done to answer this question. It was found that expertise has different effects on different Question Routing systems. The best performing expertise measure, MEC, improved the performance on NDCG of a Question Routing strategy with about 100% on the most realistic dataset. However other measures of expertise did not increase the performance of the Question Routing systems. User score decreased the performance about 10%, while a Learning to Rank approach decreased the performance up to 60%.

5.2 Discussion/Reflection

A reference process to create a Question Routing system was introduced. This reference process allowed to easily create different Question Routing system, based on different building blocks. One of these blocks was ranking the users. For this different implementations that use user expertise have been created and evaluated.

For measuring the performance of Question Routing systems care should be taken when choosing the metric to use. It was shown that due to the specific problem of Question Routing precision and MRR may not perform as well they would in other fields.

The results of the evaluation were interesting. The ranking strategy that performed the best was the MEC strategy. The maximum performance increase for the MEC strategy was 100% for the NDCG measure, while the other strategies reduced the performance.

Because of these varying results it can be concluded that user expertise can increase the performance of Question Routing systems, however care must be taken to pick the right measure for expertise. Picking the wrong measure may reduce the performance of the Question Routing system.

Different values for *intensity* were used to see how it effects the performance of Question Routing strategies. It has been shown that if there is more information available per user, the performance of the Question Routing strategy increases. However when *intensity* values increased the amount of information per users, it decreased the number of candidates as well.

5.3 Future work

The results of this work have been gathered on a single platform on a single dataset and on a single domain. In order to be able to claim generalization of the achieved results a different dataset has to be used. Stack Exchange provides dumps of all of their sites, which have the same features as Stack Overflow, but operate on different domains. This data can be used to verify that the results are cross domain. Data of a different Community Question and Answering Platform can be gathered to verify that the results will be similar to the results in this work.

This work introduces expertise for question routing. Users are routed based on their preference match with the question and their expertise. However there is no matching of question difficulty with the expertise of the user. To further improve the results of Question Routing systems question difficulty could be incorporated. An example on how this could be done is calculating the expertise of the question asker and comparing this with the expertise of the candidates.

There is quite some work done on evaluating the performance of different Question Routing systems, however there is little work done on how to properly evaluate the performance of such a system. From the results in this thesis it is clear that getting a good score is not the only performance metric that should be considered. The number of users that are selected to answer a question is important as well, I.E. the spread of the selected users. Future work could investigate which other measures of performance should be considered and how they can be used to evaluate the performance of a Question Routing system.

Finally a user study could be done to verify the performance reported in this work. Currently the actual answerers have been used as a ground truth, but they may not be a good indication of the real ground truth. Many users could have answered the question, however they did not because the question has already been answered. A user study would show how good the results of offline verification by comparing the user study with the offline verification results.

Bibliography

- [1] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 89–96, New York, NY, USA, 2005. ACM.
- [2] Shuo Chang and Aditya Pal. Routing Questions for Collaborative Answering in Community Question Answering. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, pages 494–501, New York, NY, USA, 2013. ACM.
- [3] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling . *Computer Speech & Language*, 13(4):359 – 393, 1999.
- [4] Nicole Ellison, Rebecca Heino, and Jennifer Gibbs. Managing impressions online: Self-presentation processes in the online dating environment. *Journal of Computer-Mediated Communication*, 11(2):415–441, 2006.
- [5] William A. Gale and Geoffrey Sampson. Good-turing frequency estimation without tears. *Journal of Quantitative Linguistics*, 2(3):217–237, 1995.
- [6] 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*, CIKM '08, pages 921–930, New York, NY, USA, 2008. ACM.
- [7] 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*, CSCW '12, pages 91–94, New York, NY, USA, 2012. ACM.
- [8] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22:5–53, January 2004.
- [9] Paul Jaccard. The distribution of the flora in the alpine zone.1. *New Phytologist*, 11(2):37–50, 1912.

- [10] Frederick Jelinek and Robert L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, Amsterdam, The Netherlands: North-Holland, May 1980.
- [11] Zongcheng Ji and Bin Wang. Learning to Rank for Question Routing in Community Question Answering. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management, CIKM '13*, pages 2363–2368, New York, NY, USA, 2013. ACM.
- [12] Wei-Chen Kao, Duen-Ren Liu, and Shiu-Wen Wang. Expert Finding in Question-answering Websites: A Novel Hybrid Approach. In *Proceedings of the 2010 ACM Symposium on Applied Computing, SAC '10*, pages 867–871, New York, NY, USA, 2010. ACM.
- [13] R. Kneser and H. Ney. Improved backing-off for M-gram language modeling. In *International Conference on Acoustics, Speech, and Signal Processing, 1995. ICASSP-95.*, volume 1, pages 181–184 vol.1, 1995.
- [14] W. Lam, S. Mukhopadhyay, J. Mostafa, and M. Palakal. Detection of shifts in user interests for personalized information filtering. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '96*, pages 317–325, New York, NY, USA, 1996. ACM.
- [15] Baichuan Li and Irwin King. Routing Questions to Appropriate Answerers in Community Question Answering Services. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management, CIKM '10*, pages 1585–1588, New York, NY, USA, 2010. ACM.
- [16] Baichuan Li, Irwin King, and Michael R. Lyu. Question Routing in Community Question Answering: Putting Category in Its Place. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11*, pages 2041–2044, New York, NY, USA, 2011. ACM.
- [17] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based Recommender Systems: State of the Art and Trends. In Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors, *Recommender Systems Handbook*, pages 73–105. Springer US, 2011.
- [18] Marie-Francine Moens, Juanzi Li, and Tat-Seng Chua. *Question-Answering of UGC*, chapter 9, pages 225–257. Social Media and Social Computing. Chapman and Hall, January 2014.
- [19] Martin F Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980.
- [20] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, November 1975.

- [21] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on World Wide Web, WWW '01*, pages 285–295, New York, NY, USA, 2001. ACM.
- [22] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '02*, pages 253–260, New York, NY, USA, 2002. ACM.
- [23] Ian H. Witten and T. Bell. The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *Information Theory, IEEE Transactions on*, 37(4):1085–1094, July 1991.
- [24] Jie Yang, Alessandro Bozzon, and Geert-Jan Houben. Sparrows and Owls: Characteristics of Expert Behaviour in StackOverflow. 2014.
- [25] Jun Zhang, Mark Ackerman, and Lada Adamic. Expertise Networks in Online Communities: Structure and Algorithms. In *WWW2007*.
- [26] Xiaolin Zheng, Zhongkai Hu, Aiwu Xu, Deren Chen, Kuang Liu, and Bo Li. Algorithm for recommending answer providers in community-based question answering. *J. Information Science*, 38(1):3–14, 2012.
- [27] Tom Chao Zhou, Michael R. Lyu, and Irwin King. A Classification-based Approach to Question Routing in Community Question Answering. In *Proceedings of the 21st International Conference Companion on World Wide Web, WWW '12 Companion*, pages 783–790, New York, NY, USA, 2012. ACM.
- [28] Yanhong Zhou, Gao Cong, Bin Cui, Christian S. Jensen, and Junjie Yao. Routing Questions to the Right Users in Online Communities. In *ICDE '09: Proceedings of the 2009 IEEE International Conference on Data Engineering*, pages 700–711, Washington, DC, USA, 2009. IEEE Computer Society.

Appendix A

Glossary

In this appendix an overview is given of frequently used terms and abbreviations.

A.1 Terms

user model A model that is used in the Question Routing system to create a user profile.

user profile The profile created by the Question Routing system for a specific user, based on the user model.

user properties Properties that a user has independent of a recommendation system.

question model A model that is used in the Question Routing system create a question profile.

question profile The profile created by the Question Routing system for a specific user, based on the question model.

question properties Properties that a question has independent of a recommendation system.

candidates Users that the Question Routing system can pick to route a question to.

selected candidates Users that the Question Routing system has picked and routes the question to.

A.2 Abbreviations

MEC Mean Expertise Contribution [24]

US User Score — A score given to or earned by a user on a Collaborative Question and Answering platform

CQA Collaborative Question and Answering platform

QR Question Routing

CBF Content-Based Filtering

CF Collaborative Filtering

Appendix B

Tables

Structural property	Description
Body length	The character length of the body.
Title length	The character length of the title.
Body word count	The number of white space separated tokens in the body string. Excluding HTML tags.
Title word count	The number of white space separated tokens in the title string. Excluding HTML tags.
URLs	The number of URLs used in the question.
Distinct words	The number of distinct words, another measure of length of the question.
Asker	The user id or name of the user that asked the question.

Table B.1: **The different types of STATIC statistics of a question** *that are obtained before a question is posted.*

Type	Description
Total score	The total appreciation score given by readers.
Total answer count	The total number of answers given to a question.
Total comment count	The total number of comments given to a question.
Accepted answer	If answers can be accepted by the asker of the question, this indicates if there exists an accepted answer.
Total number of views	The total number of times this question has been viewed.
Time to answer	The time it took before the first answer was provided for this question.
Time to accept	The time it took before the accepted answer was provided for this question. Only if there is an accepted answer.
Answerers	The distinct list of users that provided an answer of the question.
Total number of edits	The total number of edits a question has received.

Table B.2: **The different types of DYNAMIC statistics of a question** *that are obtained after a question is posted.*

Type	Description
Number of answers	The total number of answer given.
Number of questions	The total number of questions asked
Answer / Question ratio	The ratio of the number of answers given compared to the total number of posts done. A ratio of 1 means an only answers.
Score per answer	The total score of a user divided by the total answers
Score per post	The total score of a user divided by the questions answers and comments
Score per month	The average score per month
Expertise level	An indicator for the level of expertise

Table B.3: **The different statistics of a user.** *The statistics are chosen to reflect user interaction with the Question and Answering system.*

Set $\mathcal{A}$	Candidates	Training	Testing	Unique answerers in evaluated set (Mean)	Questions Evaluated
12/12	1054	62183	61889	9338	12386
7/12	6515	117196	123439	15530	23439
4/6	11586	130387	139414	17516	26077
2/2	25263	144529	155475	20062	28906
Set $\mathcal{B}$					
12/12	1669	137732	124302	6220	6212
7/12	10481	257873	252467	12630	12625
4/6	19007	285609	283132	14160	14163
2/2	42967	315929	317282	15806	15796

Table B.4: User and Question statistics on the filtered datasets including evaluation count
The fact that the number of candidates can be higher than the unique answerers is due to the size of the evaluated question set. Not all candidates may have provided an answer to a question in the evaluation set.

Initial	How I can read <code>DWORD</code>registry value in 64 bit machine?	
	Action	Result
1	Strip all HTML tags from the text.	How I can read DWORD registry value in 64 bit machine?
2	Strip all non alphanumeric characters.	How I can read DWORD registry value in 64 bit machine
3	Lowercase all words.	how i can read dword registry value in 64 bit machine
4	Consider words separated by a space	how i can read dword registry value in 64 bit machine
5	Remove all stopwords. ¹	read dword registry value 64 bit machine
6	Stem all words with the Porter stemming algorithm [19].	read dword registri valu 64 bit machin
7	All words shorter than 2 letters and all pure numbers are removed	read dword registri valu bit machin

Table B.5: The stages for per question processing. *After this process all words in the corpus are counted and words that only occur once or twice are removed as well.*

Appendix C

Results for set $\mathcal{A}$

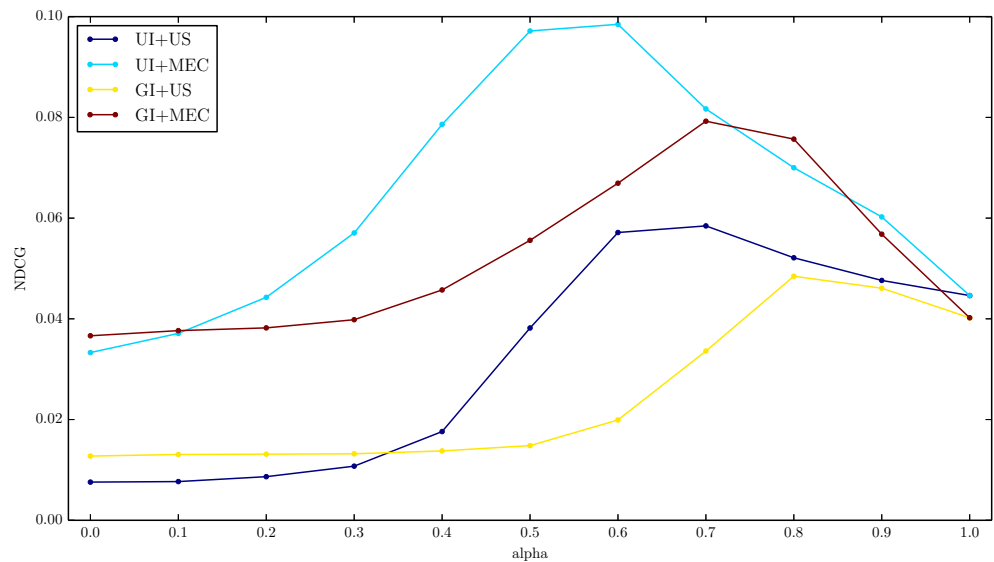


Figure C.1: Performance for different α 's The score shown is the MAP.

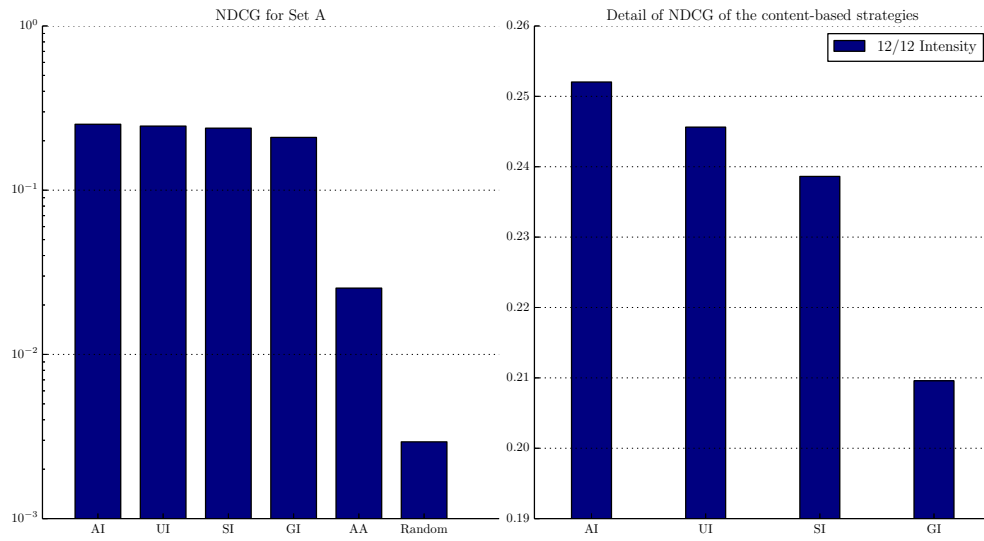


Figure C.2: Results for NDCG for a 12/12 intensity for a three month period, set $\mathcal{A}$. All content-based strategies perform an order of magnitude better than the AA approach and which in turn performs an order of magnitude better than the Random approach. The performance of content-based strategies are very close in this setting.

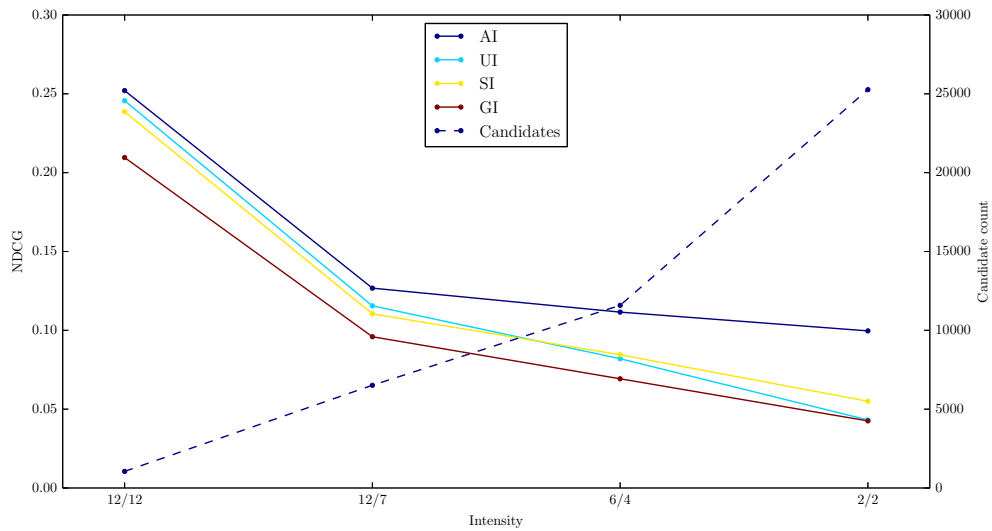


Figure C.3: The performance on NDCG@5 for the content-based matching strategies, set $\mathcal{A}$. The results show the performance on different intensities for set $\mathcal{A}$. The number of candidates is shown to provide a reference on how the data changes with each intensity.

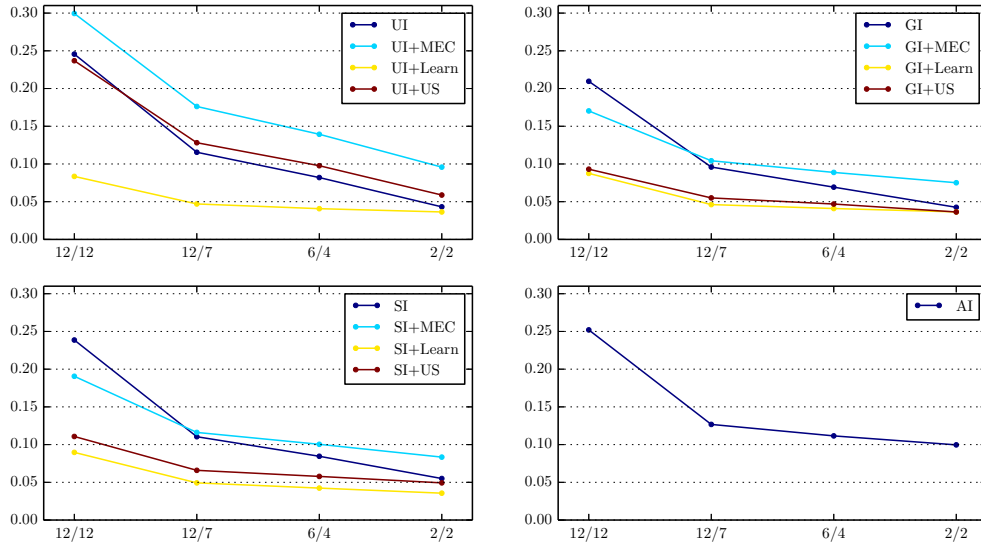


Figure C.4: Performance of different expertise ranking strategies, set $\mathcal{A}$. The labels indicate the different combinations. The first part indicates the matching strategy from Table 4.3 and the part after the + sign indicates the ranking strategy from Table 4.4.

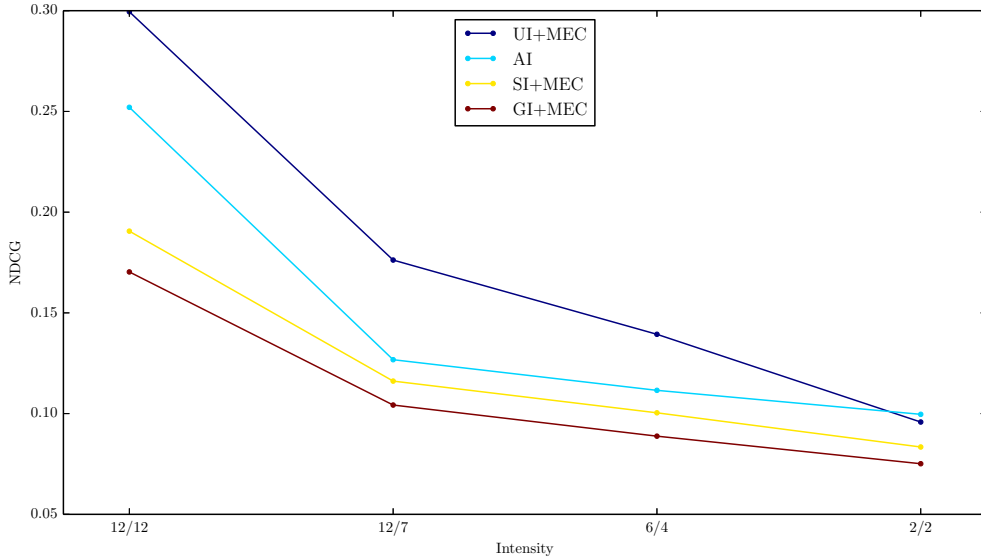
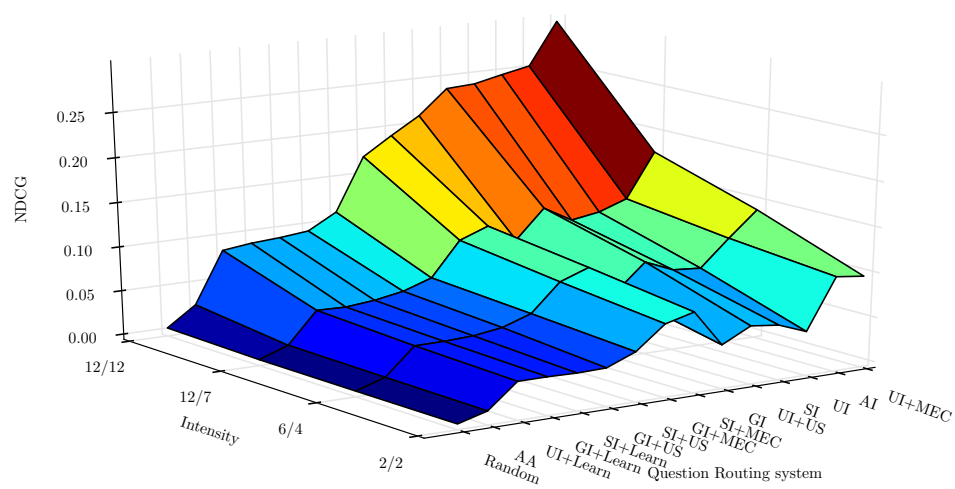


Figure C.5: The performance of MEC compared to the best performing baseline, set $\mathcal{A}$

Figure C.6: The two dimensions of testing in a single figure, set $\mathcal{A}$

Type	Reranker	P@5	%	R@5	%	F1@5	%	NDCG@5	%	MAP@5	%	MRR@1	%
Ra	—	$1.13 \cdot 10^{-03}$		$4.60 \cdot 10^{-03}$		$1.78 \cdot 10^{-03}$		$2.94 \cdot 10^{-03}$		$2.18 \cdot 10^{-03}$		$1.25 \cdot 10^{-03}$	
AA	—	$8.64 \cdot 10^{-03}$		$3.60 \cdot 10^{-02}$		$1.36 \cdot 10^{-02}$		$2.53 \cdot 10^{-02}$		$2.06 \cdot 10^{-02}$		$1.37 \cdot 10^{-02}$	
AI	—	$7.75 \cdot 10^{-02}$		$3.42 \cdot 10^{-01}$		$1.24 \cdot 10^{-01}$		$2.52 \cdot 10^{-01}$		$2.20 \cdot 10^{-01}$		$1.56 \cdot 10^{-01}$	
GI	—	$6.48 \cdot 10^{-02}$		$2.87 \cdot 10^{-01}$		$1.04 \cdot 10^{-01}$		$2.10 \cdot 10^{-01}$		$1.83 \cdot 10^{-01}$		$1.27 \cdot 10^{-01}$	
UI	—	$7.58 \cdot 10^{-02}$		$3.36 \cdot 10^{-01}$		$1.22 \cdot 10^{-01}$		$2.46 \cdot 10^{-01}$		$2.15 \cdot 10^{-01}$		$1.48 \cdot 10^{-01}$	
SI	—	$7.43 \cdot 10^{-02}$		$3.29 \cdot 10^{-01}$		$1.19 \cdot 10^{-01}$		$2.39 \cdot 10^{-01}$		$2.08 \cdot 10^{-01}$		$1.43 \cdot 10^{-01}$	
GI	MEC	$5.60 \cdot 10^{-02}$	-14	$2.45 \cdot 10^{-01}$	-15	$8.97 \cdot 10^{-02}$	-14	$1.70 \cdot 10^{-01}$	-19	$1.35 \cdot 10^{-01}$	-26	$8.44 \cdot 10^{-02}$	-34
UI	MEC	$9.08 \cdot 10^{-02}$	20	$3.95 \cdot 10^{-01}$	18	$1.45 \cdot 10^{-01}$	19	$2.99 \cdot 10^{-01}$	22	$2.56 \cdot 10^{-01}$	19	$1.89 \cdot 10^{-01}$	28
SI	MEC	$6.19 \cdot 10^{-02}$	-17	$2.73 \cdot 10^{-01}$	-17	$9.94 \cdot 10^{-02}$	-16	$1.91 \cdot 10^{-01}$	-20	$1.52 \cdot 10^{-01}$	-27	$9.59 \cdot 10^{-02}$	-33
UI	US	$7.20 \cdot 10^{-02}$	11	$3.20 \cdot 10^{-01}$	11	$1.16 \cdot 10^{-01}$	12	$2.37 \cdot 10^{-01}$	13	$2.07 \cdot 10^{-01}$	13	$1.47 \cdot 10^{-01}$	16
GI	US	$3.02 \cdot 10^{-02}$	-60	$1.36 \cdot 10^{-01}$	-60	$4.87 \cdot 10^{-02}$	-60	$9.31 \cdot 10^{-02}$	-62	$7.67 \cdot 10^{-02}$	-64	$4.70 \cdot 10^{-02}$	-68
SI	US	$3.53 \cdot 10^{-02}$	-52	$1.58 \cdot 10^{-01}$	-52	$5.70 \cdot 10^{-02}$	-52	$1.11 \cdot 10^{-01}$	-54	$9.24 \cdot 10^{-02}$	-56	$5.85 \cdot 10^{-02}$	-59
GI	Learn	$3.03 \cdot 10^{-02}$	-53	$1.29 \cdot 10^{-01}$	-55	$4.81 \cdot 10^{-02}$	-54	$8.75 \cdot 10^{-02}$	-58	$6.42 \cdot 10^{-02}$	-65	$3.82 \cdot 10^{-02}$	-70
UI	Learn	$2.90 \cdot 10^{-02}$	-62	$1.22 \cdot 10^{-01}$	-64	$4.58 \cdot 10^{-02}$	-62	$8.35 \cdot 10^{-02}$	-66	$6.05 \cdot 10^{-02}$	-72	$3.64 \cdot 10^{-02}$	-75
SI	Learn	$3.06 \cdot 10^{-02}$	-59	$1.31 \cdot 10^{-01}$	-60	$4.86 \cdot 10^{-02}$	-59	$8.97 \cdot 10^{-02}$	-62	$6.63 \cdot 10^{-02}$	-68	$4.14 \cdot 10^{-02}$	-71

Table C.1: 12/12 Intensity results for set $\mathcal{A}$

Type	Reranker	P@5	%	R@5	%	F1@5	%	NDCG@5	%	MAP@5	%	MRR@1	%
Ra	—	$2.27 \cdot 10^{-04}$		$7.26 \cdot 10^{-04}$		$3.27 \cdot 10^{-04}$		$4.77 \cdot 10^{-04}$		$3.53 \cdot 10^{-04}$		$2.57 \cdot 10^{-04}$	
AA	—	$4.40 \cdot 10^{-03}$		$1.59 \cdot 10^{-02}$		$6.60 \cdot 10^{-03}$		$1.21 \cdot 10^{-02}$		$9.16 \cdot 10^{-03}$		$6.51 \cdot 10^{-03}$	
AI	—	$4.44 \cdot 10^{-02}$		$1.72 \cdot 10^{-01}$		$6.84 \cdot 10^{-02}$		$1.27 \cdot 10^{-01}$		$1.07 \cdot 10^{-01}$		$7.67 \cdot 10^{-02}$	
GI	—	$3.33 \cdot 10^{-02}$		$1.35 \cdot 10^{-01}$		$5.21 \cdot 10^{-02}$		$9.59 \cdot 10^{-02}$		$8.07 \cdot 10^{-02}$		$5.51 \cdot 10^{-02}$	
UI	—	$4.05 \cdot 10^{-02}$		$1.64 \cdot 10^{-01}$		$6.32 \cdot 10^{-02}$		$1.16 \cdot 10^{-01}$		$9.77 \cdot 10^{-02}$		$6.56 \cdot 10^{-02}$	
SI	—	$3.88 \cdot 10^{-02}$		$1.57 \cdot 10^{-01}$		$6.05 \cdot 10^{-02}$		$1.10 \cdot 10^{-01}$		$9.30 \cdot 10^{-02}$		$6.24 \cdot 10^{-02}$	
GI	MEC	$3.67 \cdot 10^{-02}$	10	$1.36 \cdot 10^{-01}$	1	$5.58 \cdot 10^{-02}$	7	$1.04 \cdot 10^{-01}$	8	$7.59 \cdot 10^{-02}$	-6	$5.60 \cdot 10^{-02}$	2
UI	MEC	$5.95 \cdot 10^{-02}$	47	$2.27 \cdot 10^{-01}$	38	$9.13 \cdot 10^{-02}$	44	$1.76 \cdot 10^{-01}$	52	$1.41 \cdot 10^{-01}$	44	$1.10 \cdot 10^{-01}$	68
SI	MEC	$4.06 \cdot 10^{-02}$	5	$1.51 \cdot 10^{-01}$	-4	$6.16 \cdot 10^{-02}$	2	$1.16 \cdot 10^{-01}$	5	$8.62 \cdot 10^{-02}$	-7	$6.49 \cdot 10^{-02}$	4
UI	US	$4.41 \cdot 10^{-02}$	32	$1.77 \cdot 10^{-01}$	31	$6.88 \cdot 10^{-02}$	32	$1.28 \cdot 10^{-01}$	33	$1.08 \cdot 10^{-01}$	34	$7.67 \cdot 10^{-02}$	39
GI	US	$1.92 \cdot 10^{-02}$	-53	$7.80 \cdot 10^{-02}$	-52	$3.00 \cdot 10^{-02}$	-53	$5.50 \cdot 10^{-02}$	-53	$4.39 \cdot 10^{-02}$	-55	$2.78 \cdot 10^{-02}$	-58
SI	US	$2.32 \cdot 10^{-02}$	-40	$9.43 \cdot 10^{-02}$	-40	$3.62 \cdot 10^{-02}$	-40	$6.59 \cdot 10^{-02}$	-40	$5.33 \cdot 10^{-02}$	-43	$3.34 \cdot 10^{-02}$	-46
GI	Learn	$1.70 \cdot 10^{-02}$	-49	$5.51 \cdot 10^{-02}$	-59	$2.48 \cdot 10^{-02}$	-52	$4.62 \cdot 10^{-02}$	-52	$2.82 \cdot 10^{-02}$	-65	$2.42 \cdot 10^{-02}$	-56
UI	Learn	$1.73 \cdot 10^{-02}$	-57	$5.64 \cdot 10^{-02}$	-66	$2.53 \cdot 10^{-02}$	-60	$4.70 \cdot 10^{-02}$	-59	$2.90 \cdot 10^{-02}$	-70	$2.46 \cdot 10^{-02}$	-63
SI	Learn	$1.82 \cdot 10^{-02}$	-53	$5.86 \cdot 10^{-02}$	-63	$2.64 \cdot 10^{-02}$	-56	$4.93 \cdot 10^{-02}$	-55	$3.04 \cdot 10^{-02}$	-67	$2.66 \cdot 10^{-02}$	-57

Table C.2: 7/12 Intensity results for set $\mathcal{A}$

Type	Reranker	P@5	%	R@5	%	F1@5	%	NDCG@5	%	MAP@5	%	MRR@1	%
Ra	—	$7.89 \cdot 10^{-05}$		$3.01 \cdot 10^{-04}$		$1.18 \cdot 10^{-04}$		$1.55 \cdot 10^{-04}$		$1.09 \cdot 10^{-04}$		$1.79 \cdot 10^{-05}$	
AA	—	$3.87 \cdot 10^{-03}$		$1.40 \cdot 10^{-02}$		$5.81 \cdot 10^{-03}$		$1.05 \cdot 10^{-02}$		$7.97 \cdot 10^{-03}$		$5.54 \cdot 10^{-03}$	
AI	—	$3.98 \cdot 10^{-02}$		$1.50 \cdot 10^{-01}$		$6.08 \cdot 10^{-02}$		$1.12 \cdot 10^{-01}$		$9.26 \cdot 10^{-02}$		$6.79 \cdot 10^{-02}$	
GI	—	$2.49 \cdot 10^{-02}$		$9.94 \cdot 10^{-02}$		$3.87 \cdot 10^{-02}$		$6.93 \cdot 10^{-02}$		$5.74 \cdot 10^{-02}$		$3.78 \cdot 10^{-02}$	
UI	—	$2.95 \cdot 10^{-02}$		$1.18 \cdot 10^{-01}$		$4.60 \cdot 10^{-02}$		$8.20 \cdot 10^{-02}$		$6.84 \cdot 10^{-02}$		$4.49 \cdot 10^{-02}$	
SI	—	$3.06 \cdot 10^{-02}$		$1.20 \cdot 10^{-01}$		$4.73 \cdot 10^{-02}$		$8.45 \cdot 10^{-02}$		$7.06 \cdot 10^{-02}$		$4.74 \cdot 10^{-02}$	
GI	MEC	$3.21 \cdot 10^{-02}$	29	$1.13 \cdot 10^{-01}$	14	$4.79 \cdot 10^{-02}$	24	$8.88 \cdot 10^{-02}$	28	$6.23 \cdot 10^{-02}$	9	$4.87 \cdot 10^{-02}$	29
UI	MEC	$4.88 \cdot 10^{-02}$	65	$1.79 \cdot 10^{-01}$	52	$7.39 \cdot 10^{-02}$	61	$1.39 \cdot 10^{-01}$	70	$1.09 \cdot 10^{-01}$	59	$8.49 \cdot 10^{-02}$	89
SI	MEC	$3.62 \cdot 10^{-02}$	18	$1.29 \cdot 10^{-01}$	8	$5.43 \cdot 10^{-02}$	15	$1.00 \cdot 10^{-01}$	18	$7.21 \cdot 10^{-02}$	2	$5.52 \cdot 10^{-02}$	16
UI	US	$3.47 \cdot 10^{-02}$	39	$1.35 \cdot 10^{-01}$	36	$5.36 \cdot 10^{-02}$	39	$9.77 \cdot 10^{-02}$	41	$8.13 \cdot 10^{-02}$	42	$5.77 \cdot 10^{-02}$	53
GI	US	$1.69 \cdot 10^{-02}$	-43	$6.66 \cdot 10^{-02}$	-44	$2.62 \cdot 10^{-02}$	-43	$4.69 \cdot 10^{-02}$	-43	$3.69 \cdot 10^{-02}$	-46	$2.34 \cdot 10^{-02}$	-48
SI	US	$2.08 \cdot 10^{-02}$	-32	$8.19 \cdot 10^{-02}$	-32	$3.21 \cdot 10^{-02}$	-32	$5.78 \cdot 10^{-02}$	-32	$4.62 \cdot 10^{-02}$	-35	$2.98 \cdot 10^{-02}$	-37
GI	Learn	$1.54 \cdot 10^{-02}$	-38	$4.58 \cdot 10^{-02}$	-54	$2.19 \cdot 10^{-02}$	-43	$4.09 \cdot 10^{-02}$	-41	$2.36 \cdot 10^{-02}$	-59	$2.24 \cdot 10^{-02}$	-41
UI	Learn	$1.54 \cdot 10^{-02}$	-48	$4.58 \cdot 10^{-02}$	-61	$2.20 \cdot 10^{-02}$	-52	$4.07 \cdot 10^{-02}$	-50	$2.37 \cdot 10^{-02}$	-65	$2.21 \cdot 10^{-02}$	-51
SI	Learn	$1.58 \cdot 10^{-02}$	-48	$4.79 \cdot 10^{-02}$	-60	$2.26 \cdot 10^{-02}$	-52	$4.24 \cdot 10^{-02}$	-50	$2.49 \cdot 10^{-02}$	-65	$2.31 \cdot 10^{-02}$	-51

Table C.3: 4/6 Intensity results for set $\mathcal{A}$

Type	Reranker	P@5	%	R@5	%	F1@5	%	NDCG@5	%	MAP@5	%	MRR@1	%
Ra	—	$6.43 \cdot 10^{-05}$		$2.30 \cdot 10^{-04}$		$9.62 \cdot 10^{-05}$		$1.21 \cdot 10^{-04}$		$7.17 \cdot 10^{-05}$		$3.22 \cdot 10^{-05}$	
AA	—	$3.47 \cdot 10^{-03}$		$1.19 \cdot 10^{-02}$		$5.11 \cdot 10^{-03}$		$9.18 \cdot 10^{-03}$		$6.66 \cdot 10^{-03}$		$4.87 \cdot 10^{-03}$	
AI	—	$3.64 \cdot 10^{-02}$		$1.31 \cdot 10^{-01}$		$5.47 \cdot 10^{-02}$		$9.96 \cdot 10^{-02}$		$8.20 \cdot 10^{-02}$		$6.31 \cdot 10^{-02}$	
GI	—	$1.58 \cdot 10^{-02}$		$6.10 \cdot 10^{-02}$		$2.43 \cdot 10^{-02}$		$4.26 \cdot 10^{-02}$		$3.46 \cdot 10^{-02}$		$2.33 \cdot 10^{-02}$	
UI	—	$1.69 \cdot 10^{-02}$		$6.67 \cdot 10^{-02}$		$2.62 \cdot 10^{-02}$		$4.32 \cdot 10^{-02}$		$3.41 \cdot 10^{-02}$		$1.87 \cdot 10^{-02}$	
SI	—	$2.04 \cdot 10^{-02}$		$7.90 \cdot 10^{-02}$		$3.15 \cdot 10^{-02}$		$5.50 \cdot 10^{-02}$		$4.54 \cdot 10^{-02}$		$3.03 \cdot 10^{-02}$	
GI	MEC	$2.78 \cdot 10^{-02}$	76	$9.19 \cdot 10^{-02}$	51	$4.07 \cdot 10^{-02}$	67	$7.51 \cdot 10^{-02}$	76	$5.06 \cdot 10^{-02}$	46	$4.21 \cdot 10^{-02}$	81
UI	MEC	$3.52 \cdot 10^{-02}$	108	$1.22 \cdot 10^{-01}$	83	$5.24 \cdot 10^{-02}$	100	$9.58 \cdot 10^{-02}$	122	$7.11 \cdot 10^{-02}$	109	$5.61 \cdot 10^{-02}$	200
SI	MEC	$3.07 \cdot 10^{-02}$	50	$1.04 \cdot 10^{-01}$	32	$4.53 \cdot 10^{-02}$	44	$8.34 \cdot 10^{-02}$	52	$5.77 \cdot 10^{-02}$	27	$4.72 \cdot 10^{-02}$	56
UI	US	$2.25 \cdot 10^{-02}$	42	$8.63 \cdot 10^{-02}$	41	$3.46 \cdot 10^{-02}$	42	$5.88 \cdot 10^{-02}$	38	$4.68 \cdot 10^{-02}$	35	$2.92 \cdot 10^{-02}$	25
GI	US	$1.33 \cdot 10^{-02}$	-21	$5.04 \cdot 10^{-02}$	-24	$2.03 \cdot 10^{-02}$	-23	$3.64 \cdot 10^{-02}$	-16	$2.84 \cdot 10^{-02}$	-17	$1.96 \cdot 10^{-02}$	5
SI	US	$1.80 \cdot 10^{-02}$	-12	$6.88 \cdot 10^{-02}$	-13	$2.76 \cdot 10^{-02}$	-12	$4.93 \cdot 10^{-02}$	-10	$3.91 \cdot 10^{-02}$	-14	$2.63 \cdot 10^{-02}$	-13
GI	Learn	$1.40 \cdot 10^{-02}$	-11	$3.81 \cdot 10^{-02}$	-38	$1.93 \cdot 10^{-02}$	-21	$3.65 \cdot 10^{-02}$	-14	$1.97 \cdot 10^{-02}$	-43	$2.11 \cdot 10^{-02}$	-9
UI	Learn	$1.41 \cdot 10^{-02}$	-17	$3.74 \cdot 10^{-02}$	-44	$1.94 \cdot 10^{-02}$	-26	$3.64 \cdot 10^{-02}$	-16	$1.91 \cdot 10^{-02}$	-44	$2.02 \cdot 10^{-02}$	8
SI	Learn	$1.36 \cdot 10^{-02}$	-33	$3.69 \cdot 10^{-02}$	-53	$1.88 \cdot 10^{-02}$	-40	$3.56 \cdot 10^{-02}$	-35	$1.91 \cdot 10^{-02}$	-58	$2.08 \cdot 10^{-02}$	-31

Table C.4: 2/2 Intensity results for set $\mathcal{A}$

Appendix D

Results for set $\mathcal{B}$

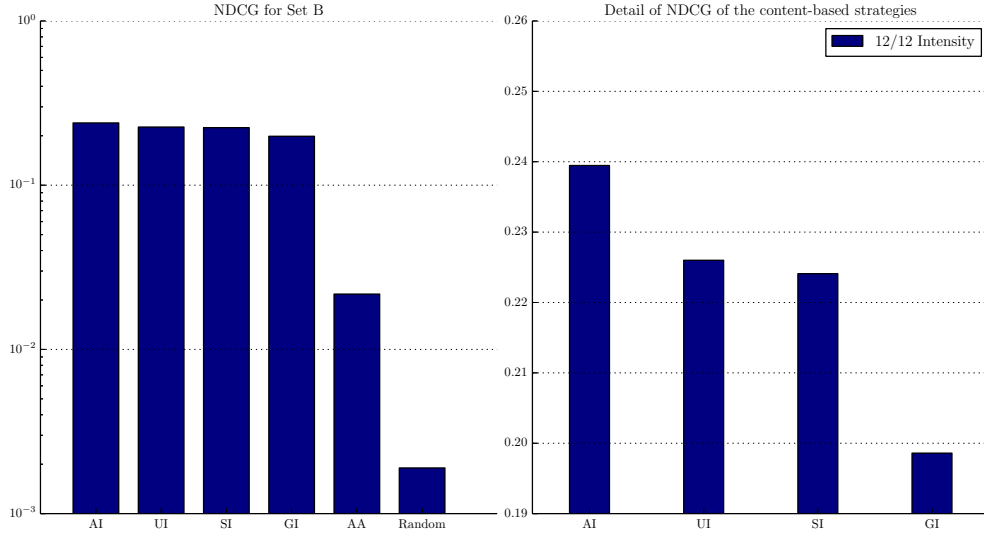


Figure D.1: **Results for NDCG for a 12/12 intensity for a six month period, set $\mathcal{B}$.** All content-based strategies perform an order of magnitude better than the AA approach and which in turn performs an order of magnitude better than the Random approach. The performance of content-based strategies are very close in this setting.

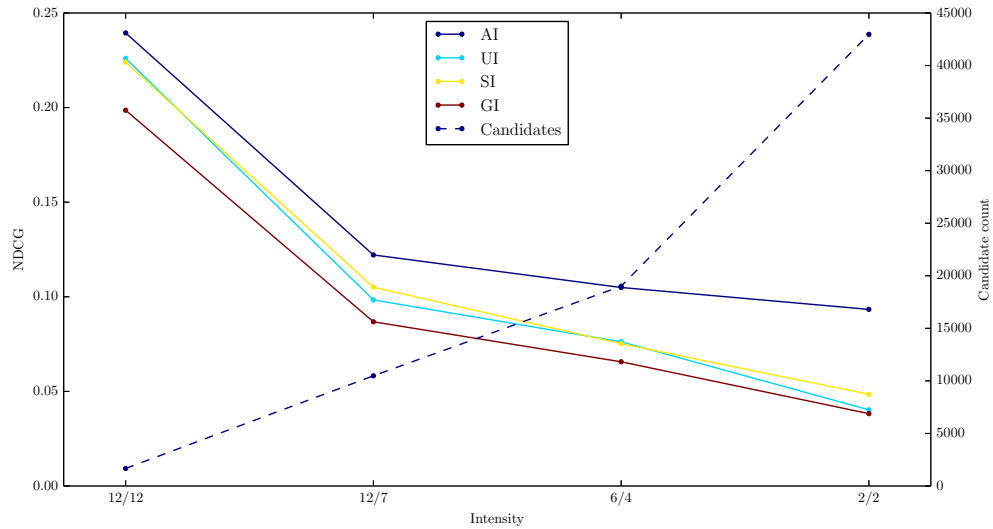


Figure D.2: **The performance on NDCG@5 for the content-based matching strategies, set $\mathcal{B}$.** The results show the performance on different intensities for set $\mathcal{B}$. The number of candidates is shown to provide a reference on how the data changes with each intensity.

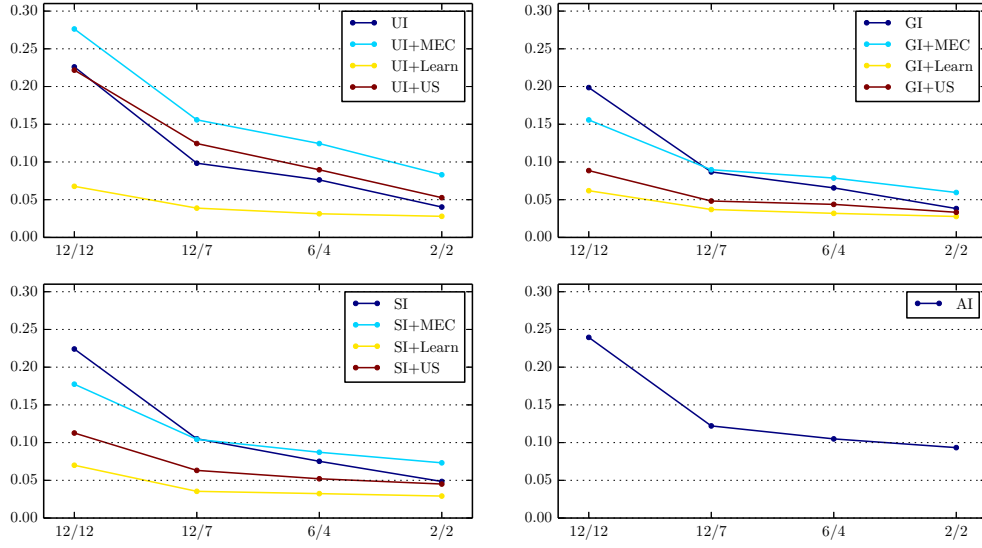


Figure D.3: **Performance of different expertise ranking strategies, set $\mathcal{B}$** The labels indicate the different combinations. The first part indicates the matching strategy from Table 4.3 and the part after the + sign indicates the ranking strategy from Table 4.4.

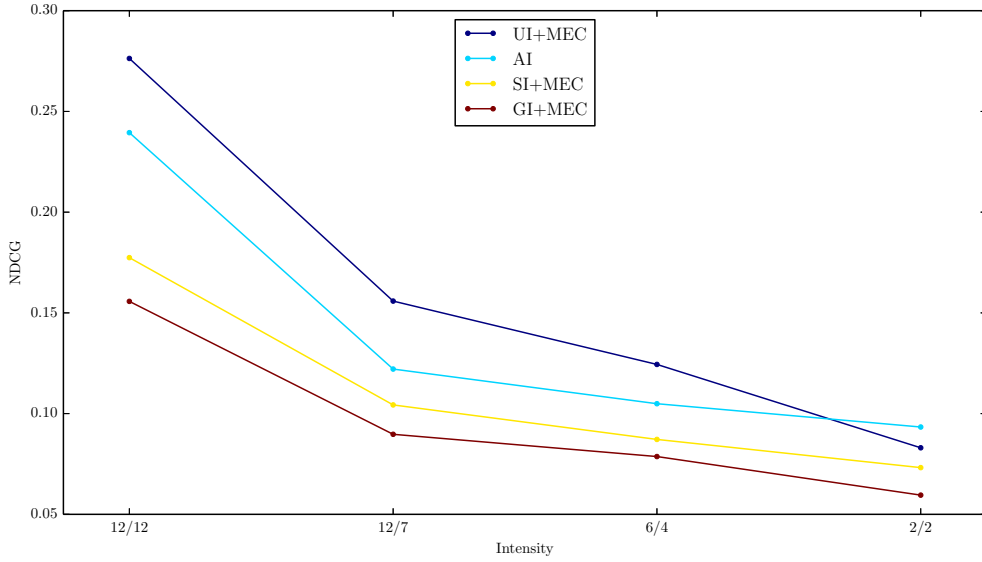
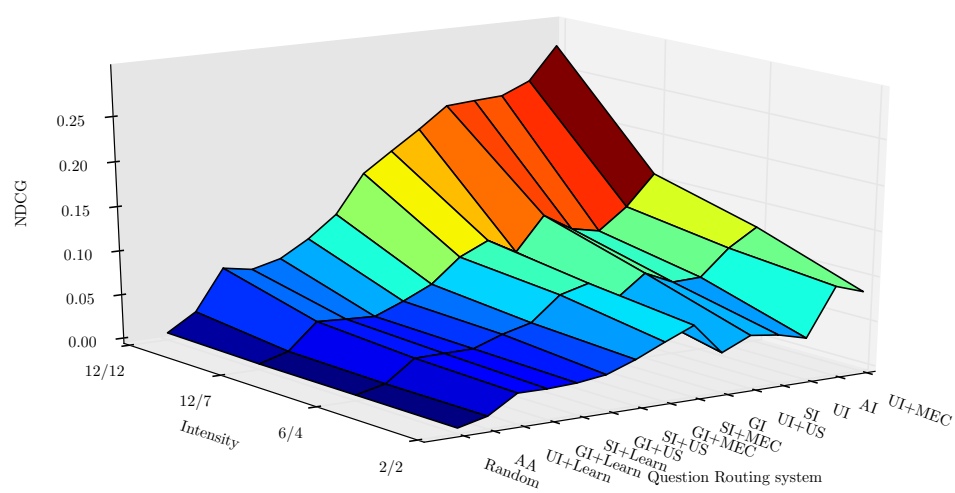


Figure D.4: **The performance of MEC compared to the best performing baseline, set $\mathcal{B}$**

Figure D.5: The two dimensions of testing in a single figure, set $\mathcal{B}$

Type	Reranker	P@5	%	R@5	%	F1@5	%	NDCG@5	%	MAP@5	%	MRR@1	%
Ra	—	$7.08 \cdot 10^{-04}$		$3.22 \cdot 10^{-03}$		$1.15 \cdot 10^{-03}$		$1.90 \cdot 10^{-03}$		$1.35 \cdot 10^{-03}$		$6.44 \cdot 10^{-04}$	
AA	—	$7.50 \cdot 10^{-03}$		$3.34 \cdot 10^{-02}$		$1.20 \cdot 10^{-02}$		$2.17 \cdot 10^{-02}$		$1.72 \cdot 10^{-02}$		$8.53 \cdot 10^{-03}$	
AI	—	$7.39 \cdot 10^{-02}$		$3.27 \cdot 10^{-01}$		$1.19 \cdot 10^{-01}$		$2.39 \cdot 10^{-01}$		$2.07 \cdot 10^{-01}$		$1.48 \cdot 10^{-01}$	
GI	—	$6.15 \cdot 10^{-02}$		$2.74 \cdot 10^{-01}$		$9.90 \cdot 10^{-02}$		$1.99 \cdot 10^{-01}$		$1.72 \cdot 10^{-01}$		$1.18 \cdot 10^{-01}$	
UI	—	$6.93 \cdot 10^{-02}$		$3.11 \cdot 10^{-01}$		$1.12 \cdot 10^{-01}$		$2.26 \cdot 10^{-01}$		$1.96 \cdot 10^{-01}$		$1.36 \cdot 10^{-01}$	
SI	—	$6.85 \cdot 10^{-02}$		$3.06 \cdot 10^{-01}$		$1.10 \cdot 10^{-01}$		$2.24 \cdot 10^{-01}$		$1.95 \cdot 10^{-01}$		$1.38 \cdot 10^{-01}$	
GI	MEC	$5.19 \cdot 10^{-02}$	-16	$2.26 \cdot 10^{-01}$	-17	$8.30 \cdot 10^{-02}$	-16	$1.56 \cdot 10^{-01}$	-22	$1.23 \cdot 10^{-01}$	-29	$7.45 \cdot 10^{-02}$	-37
UI	MEC	$8.33 \cdot 10^{-02}$	20	$3.67 \cdot 10^{-01}$	18	$1.34 \cdot 10^{-01}$	20	$2.76 \cdot 10^{-01}$	22	$2.36 \cdot 10^{-01}$	21	$1.69 \cdot 10^{-01}$	25
SI	MEC	$5.73 \cdot 10^{-02}$	-16	$2.51 \cdot 10^{-01}$	-18	$9.19 \cdot 10^{-02}$	-17	$1.77 \cdot 10^{-01}$	-21	$1.43 \cdot 10^{-01}$	-27	$9.29 \cdot 10^{-02}$	-32
GI	US	$2.90 \cdot 10^{-02}$	-53	$1.32 \cdot 10^{-01}$	-52	$4.69 \cdot 10^{-02}$	-53	$8.86 \cdot 10^{-02}$	-55	$7.27 \cdot 10^{-02}$	-58	$3.96 \cdot 10^{-02}$	-67
UI	US	$6.63 \cdot 10^{-02}$	-4	$2.98 \cdot 10^{-01}$	-4	$1.07 \cdot 10^{-01}$	-4	$2.22 \cdot 10^{-01}$	-2	$1.95 \cdot 10^{-01}$	0	$1.40 \cdot 10^{-01}$	3
SI	US	$3.62 \cdot 10^{-02}$	-47	$1.63 \cdot 10^{-01}$	-47	$5.85 \cdot 10^{-02}$	-47	$1.13 \cdot 10^{-01}$	-50	$9.29 \cdot 10^{-02}$	-52	$5.48 \cdot 10^{-02}$	-60
GI	Learn	$2.19 \cdot 10^{-02}$	-64	$9.17 \cdot 10^{-02}$	-67	$3.46 \cdot 10^{-02}$	-65	$6.19 \cdot 10^{-02}$	-69	$4.46 \cdot 10^{-02}$	-74	$2.57 \cdot 10^{-02}$	-78
UI	Learn	$2.34 \cdot 10^{-02}$	-66	$9.84 \cdot 10^{-02}$	-68	$3.70 \cdot 10^{-02}$	-67	$6.77 \cdot 10^{-02}$	-70	$4.98 \cdot 10^{-02}$	-75	$3.05 \cdot 10^{-02}$	-78
SI	Learn	$2.43 \cdot 10^{-02}$	-65	$1.02 \cdot 10^{-01}$	-67	$3.84 \cdot 10^{-02}$	-65	$7.00 \cdot 10^{-02}$	-69	$5.11 \cdot 10^{-02}$	-74	$3.20 \cdot 10^{-02}$	-77

Table D.1: 12/12 Intensity results for set $\mathcal{B}$

Type	Reranker	P@5	%	R@5	%	F1@5	%	NDCG@5	%	MAP@5	%	MRR@1	%
Ra	—	$6.34 \cdot 10^{-05}$		$3.17 \cdot 10^{-04}$		$1.06 \cdot 10^{-04}$		$1.58 \cdot 10^{-04}$		$1.06 \cdot 10^{-04}$		$0.00 \cdot 10^{+00}$	
AA	—	$3.56 \cdot 10^{-03}$		$1.40 \cdot 10^{-02}$		$5.52 \cdot 10^{-03}$		$9.61 \cdot 10^{-03}$		$6.72 \cdot 10^{-03}$		$4.44 \cdot 10^{-03}$	
AI	—	$4.13 \cdot 10^{-02}$		$1.64 \cdot 10^{-01}$		$6.41 \cdot 10^{-02}$		$1.22 \cdot 10^{-01}$		$1.04 \cdot 10^{-01}$		$7.74 \cdot 10^{-02}$	
GI	—	$3.05 \cdot 10^{-02}$		$1.24 \cdot 10^{-01}$		$4.76 \cdot 10^{-02}$		$8.68 \cdot 10^{-02}$		$7.23 \cdot 10^{-02}$		$4.79 \cdot 10^{-02}$	
UI	—	$3.39 \cdot 10^{-02}$		$1.39 \cdot 10^{-01}$		$5.32 \cdot 10^{-02}$		$9.84 \cdot 10^{-02}$		$8.30 \cdot 10^{-02}$		$5.51 \cdot 10^{-02}$	
SI	—	$3.63 \cdot 10^{-02}$		$1.48 \cdot 10^{-01}$		$5.69 \cdot 10^{-02}$		$1.05 \cdot 10^{-01}$		$8.85 \cdot 10^{-02}$		$6.05 \cdot 10^{-02}$	
GI	MEC	$3.18 \cdot 10^{-02}$	4	$1.19 \cdot 10^{-01}$	-4	$4.84 \cdot 10^{-02}$	2	$8.97 \cdot 10^{-02}$	3	$6.58 \cdot 10^{-02}$	-9	$4.77 \cdot 10^{-02}$	-1
UI	MEC	$5.19 \cdot 10^{-02}$	53	$2.00 \cdot 10^{-01}$	44	$8.00 \cdot 10^{-02}$	50	$1.56 \cdot 10^{-01}$	58	$1.25 \cdot 10^{-01}$	51	$9.65 \cdot 10^{-02}$	75
SI	MEC	$3.63 \cdot 10^{-02}$	0	$1.38 \cdot 10^{-01}$	-7	$5.56 \cdot 10^{-02}$	-2	$1.04 \cdot 10^{-01}$	-1	$7.80 \cdot 10^{-02}$	-12	$5.57 \cdot 10^{-02}$	-8
GI	US	$1.77 \cdot 10^{-02}$	-42	$7.20 \cdot 10^{-02}$	-42	$2.77 \cdot 10^{-02}$	-42	$4.82 \cdot 10^{-02}$	-44	$3.81 \cdot 10^{-02}$	-47	$2.22 \cdot 10^{-02}$	-54
UI	US	$4.24 \cdot 10^{-02}$	25	$1.71 \cdot 10^{-01}$	23	$6.63 \cdot 10^{-02}$	25	$1.24 \cdot 10^{-01}$	27	$1.05 \cdot 10^{-01}$	26	$7.28 \cdot 10^{-02}$	32
SI	US	$2.24 \cdot 10^{-02}$	-38	$9.12 \cdot 10^{-02}$	-39	$3.50 \cdot 10^{-02}$	-38	$6.32 \cdot 10^{-02}$	-40	$5.15 \cdot 10^{-02}$	-42	$3.20 \cdot 10^{-02}$	-47
GI	Learn	$1.39 \cdot 10^{-02}$	-54	$4.43 \cdot 10^{-02}$	-64	$2.03 \cdot 10^{-02}$	-57	$3.70 \cdot 10^{-02}$	-57	$2.25 \cdot 10^{-02}$	-69	$1.94 \cdot 10^{-02}$	-60
UI	Learn	$1.44 \cdot 10^{-02}$	-57	$4.73 \cdot 10^{-02}$	-66	$2.12 \cdot 10^{-02}$	-60	$3.88 \cdot 10^{-02}$	-61	$2.39 \cdot 10^{-02}$	-71	$1.97 \cdot 10^{-02}$	-64
SI	Learn	$1.33 \cdot 10^{-02}$	-63	$4.36 \cdot 10^{-02}$	-71	$1.95 \cdot 10^{-02}$	-66	$3.54 \cdot 10^{-02}$	-66	$2.12 \cdot 10^{-02}$	-76	$1.75 \cdot 10^{-02}$	-71

Table D.2: 7/12 Intensity results for set $\mathcal{B}$

Type	Reranker	P@5	%	R@5	%	F1@5	%	NDCG@5	%	MAP@5	%	MRR@1	%
Ra	—	$8.48 \cdot 10^{-05}$		$2.85 \cdot 10^{-04}$		$1.23 \cdot 10^{-04}$		$1.41 \cdot 10^{-04}$		$8.36 \cdot 10^{-05}$		0.00E+00	
AA	—	$2.97 \cdot 10^{-03}$		$1.10 \cdot 10^{-02}$		$4.50 \cdot 10^{-03}$		$7.70 \cdot 10^{-03}$		$5.31 \cdot 10^{-03}$		$4.16 \cdot 10^{-03}$	
AI	—	$3.65 \cdot 10^{-02}$		$1.40 \cdot 10^{-01}$		$5.60 \cdot 10^{-02}$		$1.05 \cdot 10^{-01}$		$8.69 \cdot 10^{-02}$		$6.64 \cdot 10^{-02}$	
GI	—	$2.34 \cdot 10^{-02}$		$9.35 \cdot 10^{-02}$		$3.65 \cdot 10^{-02}$		$6.56 \cdot 10^{-02}$		$5.37 \cdot 10^{-02}$		$3.58 \cdot 10^{-02}$	
UI	—	$2.72 \cdot 10^{-02}$		$1.09 \cdot 10^{-01}$		$4.24 \cdot 10^{-02}$		$7.62 \cdot 10^{-02}$		$6.39 \cdot 10^{-02}$		$4.25 \cdot 10^{-02}$	
SI	—	$2.69 \cdot 10^{-02}$		$1.06 \cdot 10^{-01}$		$4.16 \cdot 10^{-02}$		$7.52 \cdot 10^{-02}$		$6.24 \cdot 10^{-02}$		$4.23 \cdot 10^{-02}$	
GI	MEC	$2.83 \cdot 10^{-02}$	21	$1.02 \cdot 10^{-01}$	9	$4.26 \cdot 10^{-02}$	17	$7.87 \cdot 10^{-02}$	20	$5.58 \cdot 10^{-02}$	4	$4.18 \cdot 10^{-02}$	17
UI	MEC	$4.38 \cdot 10^{-02}$	61	$1.63 \cdot 10^{-01}$	49	$6.66 \cdot 10^{-02}$	57	$1.24 \cdot 10^{-01}$	63	$9.71 \cdot 10^{-02}$	52	$7.31 \cdot 10^{-02}$	72
SI	MEC	$3.16 \cdot 10^{-02}$	18	$1.14 \cdot 10^{-01}$	7	$4.76 \cdot 10^{-02}$	14	$8.72 \cdot 10^{-02}$	16	$6.25 \cdot 10^{-02}$	0	$4.63 \cdot 10^{-02}$	9
GI	US	$1.57 \cdot 10^{-02}$	-33	$6.33 \cdot 10^{-02}$	-32	$2.45 \cdot 10^{-02}$	-33	$4.38 \cdot 10^{-02}$	-33	$3.48 \cdot 10^{-02}$	-35	$2.13 \cdot 10^{-02}$	-41
UI	US	$3.20 \cdot 10^{-02}$	18	$1.27 \cdot 10^{-01}$	16	$4.97 \cdot 10^{-02}$	17	$8.96 \cdot 10^{-02}$	18	$7.35 \cdot 10^{-02}$	15	$4.94 \cdot 10^{-02}$	16
SI	US	$1.86 \cdot 10^{-02}$	-31	$7.35 \cdot 10^{-02}$	-31	$2.88 \cdot 10^{-02}$	-31	$5.20 \cdot 10^{-02}$	-31	$4.20 \cdot 10^{-02}$	-33	$2.75 \cdot 10^{-02}$	-35
GI	Learn	$1.22 \cdot 10^{-02}$	-48	$3.59 \cdot 10^{-02}$	-62	$1.74 \cdot 10^{-02}$	-52	$3.19 \cdot 10^{-02}$	-51	$1.80 \cdot 10^{-02}$	-66	$1.65 \cdot 10^{-02}$	-54
UI	Learn	$1.21 \cdot 10^{-02}$	-56	$3.51 \cdot 10^{-02}$	-68	$1.70 \cdot 10^{-02}$	-60	$3.13 \cdot 10^{-02}$	-59	$1.77 \cdot 10^{-02}$	-72	$1.61 \cdot 10^{-02}$	-62
SI	Learn	$1.23 \cdot 10^{-02}$	-54	$3.64 \cdot 10^{-02}$	-66	$1.75 \cdot 10^{-02}$	-58	$3.24 \cdot 10^{-02}$	-57	$1.87 \cdot 10^{-02}$	-70	$1.67 \cdot 10^{-02}$	-60

Table D.3: 4/6 Intensity results for set $\mathcal{B}$

Type	Reranker	P@5	%	R@5	%	F1@5	%	NDCG@5	%	MAP@5	%	MRR@1	%
Ra	—	$2.52 \cdot 10^{-05}$		$4.41 \cdot 10^{-05}$		$3.06 \cdot 10^{-05}$		$8.17 \cdot 10^{-06}$		$1.21 \cdot 10^{-05}$		0.00E+00	
AA	—	$2.92 \cdot 10^{-03}$		$1.09 \cdot 10^{-02}$		$4.43 \cdot 10^{-03}$		$7.83 \cdot 10^{-03}$		$5.55 \cdot 10^{-03}$		$3.98 \cdot 10^{-03}$	
AI	—	$3.34 \cdot 10^{-02}$		$1.23 \cdot 10^{-01}$		$5.06 \cdot 10^{-02}$		$9.33 \cdot 10^{-02}$		$7.64 \cdot 10^{-02}$		$5.93 \cdot 10^{-02}$	
GI	—	$1.44 \cdot 10^{-02}$		$5.53 \cdot 10^{-02}$		$2.21 \cdot 10^{-02}$		$3.82 \cdot 10^{-02}$		$3.09 \cdot 10^{-02}$		$1.97 \cdot 10^{-02}$	
UI	—	$1.49 \cdot 10^{-02}$		$6.00 \cdot 10^{-02}$		$2.33 \cdot 10^{-02}$		$4.02 \cdot 10^{-02}$		$3.24 \cdot 10^{-02}$		$1.93 \cdot 10^{-02}$	
SI	—	$1.80 \cdot 10^{-02}$		$6.84 \cdot 10^{-02}$		$2.75 \cdot 10^{-02}$		$4.85 \cdot 10^{-02}$		$3.93 \cdot 10^{-02}$		$2.68 \cdot 10^{-02}$	
GI	MEC	$2.26 \cdot 10^{-02}$	57	$7.50 \cdot 10^{-02}$	36	$3.31 \cdot 10^{-02}$	50	$5.95 \cdot 10^{-02}$	56	$3.98 \cdot 10^{-02}$	29	$3.07 \cdot 10^{-02}$	56
UI	MEC	$3.09 \cdot 10^{-02}$	107	$1.08 \cdot 10^{-01}$	80	$4.61 \cdot 10^{-02}$	98	$8.30 \cdot 10^{-02}$	106	$6.14 \cdot 10^{-02}$	90	$4.60 \cdot 10^{-02}$	138
SI	MEC	$2.73 \cdot 10^{-02}$	52	$9.18 \cdot 10^{-02}$	34	$4.02 \cdot 10^{-02}$	46	$7.32 \cdot 10^{-02}$	51	$5.11 \cdot 10^{-02}$	30	$4.07 \cdot 10^{-02}$	52
GI	US	$1.26 \cdot 10^{-02}$	-13	$4.78 \cdot 10^{-02}$	-14	$1.92 \cdot 10^{-02}$	-13	$3.33 \cdot 10^{-02}$	-13	$2.58 \cdot 10^{-02}$	-17	$1.64 \cdot 10^{-02}$	-17
UI	US	$1.95 \cdot 10^{-02}$	31	$7.51 \cdot 10^{-02}$	25	$3.00 \cdot 10^{-02}$	29	$5.27 \cdot 10^{-02}$	31	$4.27 \cdot 10^{-02}$	32	$2.86 \cdot 10^{-02}$	48
SI	US	$1.63 \cdot 10^{-02}$	-9	$6.25 \cdot 10^{-02}$	-9	$2.51 \cdot 10^{-02}$	-9	$4.51 \cdot 10^{-02}$	-7	$3.57 \cdot 10^{-02}$	-9	$2.44 \cdot 10^{-02}$	-9
GI	Learn	$1.06 \cdot 10^{-02}$	-26	$2.79 \cdot 10^{-02}$	-50	$1.45 \cdot 10^{-02}$	-34	$2.77 \cdot 10^{-02}$	-28	$1.43 \cdot 10^{-02}$	-54	$1.63 \cdot 10^{-02}$	-17
UI	Learn	$1.10 \cdot 10^{-02}$	-26	$2.87 \cdot 10^{-02}$	-52	$1.50 \cdot 10^{-02}$	-36	$2.79 \cdot 10^{-02}$	-31	$1.45 \cdot 10^{-02}$	-55	$1.53 \cdot 10^{-02}$	-21
SI	Learn	$1.14 \cdot 10^{-02}$	-36	$3.05 \cdot 10^{-02}$	-55	$1.57 \cdot 10^{-02}$	-43	$2.91 \cdot 10^{-02}$	-40	$1.54 \cdot 10^{-02}$	-61	$1.51 \cdot 10^{-02}$	-44

Table D.4: 2/2 Intensity results for set $\mathcal{B}$