Semantic-Enhanced Training Data Augmentation Methods for Long-Tail Entity Recognition Models

Mesbah, Sepideh

DOI
10.4233/uuid:dbbfe1fc-bf63-45f0-8cf2-28ed7dab90eb

Publication date
2020

Document Version
Final published version

Citation (APA)
Mesbah, S. (2020). *Semantic-Enhanced Training Data Augmentation Methods for Long-Tail Entity Recognition Models*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:dbbfe1fc-bf63-45f0-8cf2-28ed7dab90eb

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Semantic-Enhanced Training Data Augmentation Methods for Long-Tail Entity Recognition Models

Sepideh Mesbah

Semantic-Enhanced Training Data Augmentation Methods for Long-Tail Entity Recognition Models

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, Prof.dr.ir. T.H.J.J. van der Hagen,
Chair of the Board for Doctorates
to be defended publicly on
Wednesday 20 May 2020 at 12:30 o'clock

by

Sepideh MESBAH

Master of Science in Informatics, Technical University of Munich, Germany
born in Tehran, Iran.

This dissertation has been approved by the promotors.

Promotor: prof. dr. ir. G.J.P.M Houben
Promotor: prof. dr. ir. A. Bozzon
Copromotor: dr. C. Lofi

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. ir. G.J.P.M Houben	Delft University of Technology, promotor
Prof. dr. ir. A. Bozzon	Delft University of Technology, promotor
Dr. C. Lofi	Delft University of Technology, copromotor

Independent members:

Prof. dr. P. Fraternali	Politecnico di Milano, Italy
Prof. dr. W.-T. Balke	Technische Universität Braunschweig, Germany
Prof. dr. A.P.J. van den Bosch	Tilburg University
Prof. dr. A. Van Deursen	Delft University of Technology

SIKS Dissertation Series No.
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

Published and distributed by: Sepideh Mesbah
E-mail: mesbah.s@gmail.com

Keywords: Long-tail Name Entity Recognition, Training Data Augmentation, Semantic Enrichment

Printing and cover design by: ProefschriftMaken

ISBN: 978-94-6380-808-8

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/.`

Contents

Acknowledgements

This PhD thesis is the output of the effort and support of several people to whom I am extremely grateful. First of all, I wish to express my deepest gratitude to my supervisory team Geert-Jan Houben, Alessandro Bozzon and Christoph Lofi for giving me the opportunity to work under their supervision. I am very thankful for your continuous support, encouragement, and kindness. Geert-Jan, thank you for giving me the opportunity to start this Ph.D. and supporting me throughout these four years. Thanks for your critical and constructive comments and for always making me think about the bigger picture. You are a great source of inspiration for me. Alessandro, thank you for your full support throughout this journey, you always gave me good guidance, Im grateful for your support in my development as a researcher. Thank you for supporting me in articulating my thoughts into meaningful outcomes. Christoph, our meetings and discussions have always been lively as well as challenging and your advice has helped me tremendously in completing this work. Thanks for giving useful advice in shaping the narrative and writing up the papers and the thesis.

I would like to thank the members of my defense committee: Prof. Arie Van Deursen, Prof. Piero Fraternali, Prof. Antal Van Den Bosch, Prof. Tilo Balke and Prof. Andy Zaidman, for accepting to be part of my committee and providing me with valuable feedback on this thesis.

Many thanks to the current and former members of the Web information systems group at Delft University of Technology: Marcus Specht, Claudia Hauff, Asterios Katsifodimos, Nava Tintarev, Andrea Mauri, Achilleas Psyllidis, Dimitrios Bountouridis, Panagiotis Mavridis, Shabnam Najafian, Arthur Camara, Dan Davis, Vincent Gong, Christos Koutras, Felipe Moraes, Jasper Oosterman, Jie Yang, Gustavo Penha, Ioannis Petros Samiotis, Sihang Qiu, Yue Zhao, Carlo van der Valk, Pavel Kucherbaev, Ujwal Gadiraju, Marios Fragkoulis, Oana Inel, Ioana jivet, David Maxwell, Esther tan, Agathe balayn, Tim Draws, Vincent Gong, Yoon Lee, Ziyu Li, Nesse van der Meer, Nirmal Roy, Sara Salimzadeh, Georgios Siachamis, Peide Zhu, Mesut Kaya, Pedro Fortunato Silvestre Manuel Valle Torre, Guanliang Chen, Andra Lonescu, Tarmo Robal, Mohammad Khalil, Mónica Marrero, Tamara Brusik, Roniet Sharabi and Daphne Stephan. Special thanks to Jie for the valuable brainstorming sessions and guidance. To Guanliang for helping me throughout the difficult times and in the final stage of my PhD journey even from Australia. To Shabnam for the wonderful talks and great breaks we had. To my amazing office mates Sihang, Petros, Felipe, Christos and Agathe for the great moments we

shared.

I had the privilege to work with a number of talented collaborators: Sara Bashirieh, Kyriakos Frageskos, Manuel Valle Torre, Daniel Vliegenthart, Tom Harting, Manolis Manousogiannis, Robert-Jan Sips, Zoltan Szlavik, Selene Baez Santamaria. Thanks for the great collaborations and results. Special thanks to Robert-Jan who gave me the opportunity to do an internship at Mytomorrows.

I really appreciate spending time with my dear friends, Sara, Aynaz, Assal, Shabnam, Samira, Soheila, Peyman, Jos, Kaveh and Siamak. Thank you for the laughs and good times. You guys did a great job in keeping me a happy Sepideh.

This journey would not have been possible without the love and support of my family. My Parents (Baba Maman) thanks for loving me unconditionally and for showing me the meaning of patience and perseverance. Thanks for supporting me in everything I do and pushing me farther then I thought I could go. My brother Ali and sister-in-law Negin, thank you for your caring, guidance and encouragement. Ali, you were the first one who thought me how to write a paper. My sister Rahele and brother-in-law Parham thank you for always being there for me and helping me through all those tough times. Thanks for the great weekends, it helped me boost up my energy. Rahele thanks for helping me to become a stronger person emotionally. Thanks to my little nephews Daniel, Adrian and my niece Danica, all the free cuddles and kisses you offered me have been the best PhD therapy. I would like to thank my in-laws Dayan, Babak, Parisa, Ghazale, Ramin, Afshin, Ghazal, Baran, Hasti for their care, smiles, and warmness. Last, but certainly not least, I am indebted to my best friend and husband, Shahin. I feel so fortunate you were by my side supporting me in every step of this challenging path. Thank you for cheering me up in my vulnerable moments and for always seeing the positive side of me.

I am so blessed to have all of you in my life. Thank you for everything.

Chapter 1

Introduction

Named Entity Recognition (NER) is a basic Information Extraction task that can be formulated as a sequence labeling problem which assigns a named entity type to each word in the input sequence. NER was originally focused on only recognizing proper name mentions such as person, location, and organization. It later expanded to the task of identifying and typing words or phrases in a text that refers to certain classes of interest (e.g., disease, Adverse Drug Reactions) [150, 15, 44]. In this thesis, we will be using this more general definition of NER. NER enables a wide range of natural language applications such as question answering [178], automatic content summarization [163], machine translation [32], semantic search [80] and ontology population [59, 224]. Approaches to NER differ [73] and they are based on techniques that are dictionary-based [174], rule-based [69, 175, 195], machine learning-based [192, 12, 7, 24] or hybrid-based (combination of rule-based and machine learning) [205, 117]. This thesis, focuses on machine learning-based NER techniques. We emphasize the problem of the lack of training data, arguably the largest bottleneck in training machine learning-based NER techniques.

1.1 Problem Statement

Machine learning-based NER techniques [12, 7, 215] have shown to achieve an impressive performance (e.g., F-score up to 93.5[1]) in the case of entities (e.g., locations, organizations, dates) for which a large amount of human-labeled training data is available. However, these techniques show their limits when it comes to long-tail entities [177].
Long-tail entities are entities that have a low frequency in the document collections and usually have no reference to existing Knowledge Bases [58]. Long-tail entities are usually relevant in specific usage contexts, implied by a domain, time, topic, or community [93]. For instance, in science, domain-specific entities are long-tail entities that are restricted to a given domain such as biomedical science, data science, or history. These domain-specific entities often appear in scientific publications and play a crucial role in understanding

[1] A repository to keep track of the progress in Natural Language Processing (NLP): http://nlpprogress.com/english/named_entity_recognition.html

the semantics of the scientific texts. Table 1.1 highlights examples of long-tail entities in scientific text: *SimFusion+*[2] and *WebKB*[3] are entities of types *Method* (i.e., an algorithm) and *Dataset*.

As another example, user-generated phrases are examples of long-tail entities present in User Generated Content (UGC) published in online communication platforms such as Twitter or Reddit. User-generated phrases are rare; they exhibit linguistic differences across different online communities (i.e., Twitter and AskAPatient); and they convey a given concept using diverse expressions and ambiguous mentions. This makes the user-generated phrases challenging to be recognized automatically. As shown in Table 1.1, the phrases *"No sleep"* and *"can't fall asleep"* are of type *"Adverse Drug Reaction"* and refer to the concept *"Insomnia"*[4]. Devising techniques to automatically detect and type long-tail phrases [5] in User Generated Content can provide valuable insights for monitoring public health, marketing, etc.

Table 1.1: Examples of long-tail entities (in bold) in different sources.

Scientific Publication	We evaluated the performance of **SimFusion+** on the **WebKB** dataset
AskAPatient (UGC)	I took evista for the first time about 15 years ago. It was the worst year of my life. **No sleep** and constant **night sweats**
Twitter (UGC)	Exhausted...**can't fall asleep**. Don't wanna take a trazadone and wake up hungover. #Sleepdisorderproblems

State-of-the-art Named Entity Recognition (NER) methods [23, 101, 49] require human-labeled training datasets for their supervised machine learning. These datasets are expensive and time-consuming to obtain for long-tail entities. A cheaper alternative is to generate labeled training data by retrieving existing instances of the targeted entity type from Knowledge Bases (KBs) [23]. This of course requires that the desired entity type is well-covered in the KB. In recent years, data augmentation has become a popular technique for automatically increasing the size of labeled training data [176, 41]. Studies [82, 108, 176] have shown that data augmentation can improve the performance of machine learning-based techniques by automatically expanding the size of labeled training samples and representing a more comprehensive set of possible data points.

In this thesis, we focus on the specific problem of training data augmentation and investigate how different training data augmentation techniques can improve the performance of NER models. Figure 1.1 provides a high-level overview of the pursued approach. Our intuition is that by exploring the implicit semantics and structure of a seed labeled training data set, as well as the unlabeled data in the domain of interest, we can obtain larger

[2] SimFusion+ is an algorithm for measuring similarity between objects in a web graph

[3] WebKB consists of web pages and hyperlinks from different computer science departments

[4] https://www.med.upenn.edu/ocrobjects/PM/2_glossary.of.lay.terms.pdf

[5] long-tail phrase detection and typing is also the subclass of the sequence labeling problem, which instead of detecting and typing of only named entities, focuses on recognizing phrases/sequence of words

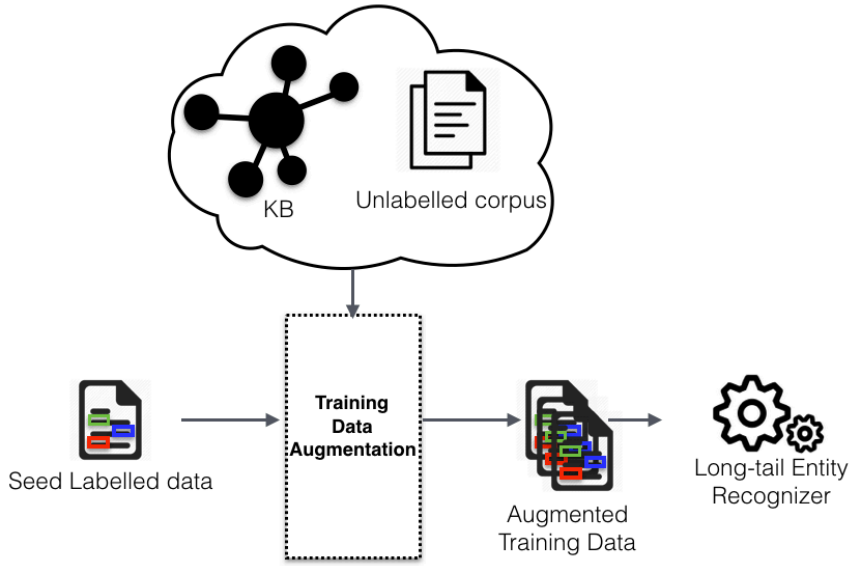


Figure 1.1: Overview of the focus of the thesis: semantic-enhanced methods to augment training data for improving the supervised training of long-tail Entity Recognition

amount of good quality training data. A NER model can then use this data to achieve satisfactory recognition and typing performance.

1.2 Research Questions

This thesis investigates the following main research question:

- **MRQ**: How can we augment training data to improve the supervised training of Long-tail Entity Recognition (L-tER) algorithms?

To answer our main research question, we organized the work in four research sub-questions, where we investigate techniques used to support the extraction and typing of long-tail entities contained in scientific publications (RQ 1, RQ 2, RQ 3) and User Generated Content (RQ 4). We start by using a state-of-the-art pre-trained NER to check if it can be used for extracting the long-tail entities (RQ 1). The results show that generic NER is not suitable for long-tail Entity Recognition and new models need to be trained. The lack of training data is the largest bottleneck in long-tail Entity Recognition **(L-tER)** training. We tackle this problem by augmenting the training datasets by enhancing their size using semantic expansion techniques (RQ2) and generative models (RQ 4); and by improving their quality using collaborative feedback from users (RQ 3).

In RQ 2 we enhance the size of the training data using semantic expansion and heuristic techniques. As these heuristics are prone to failure, the overall achievable performance

is limited. In RQ 3, we therefore introduce a collaborative approach which incrementally incorporates human feedback on the relevance of extracted entities into the training cycle. We further continue our research by focusing on supporting the extraction and typing of user-generated phrases that appear in ungrammatical sentence structures and non-standard words, in contrast to the text of scientific publications which are structured. This helps us to further our understanding of how to support the supervised training of Long-tail Entity Recognizer (L-tER) in different sources with different properties. To this end, we devise a technique for augmenting the training data using deep generative models (RQ 4).

Our first research question can be formulated as follows:

- **RQ 1:** To what extent can pre-trained NER recognize long-tail entities?

  Pre-trained NER is trained on large amounts of training data to recognize generic entity types (e.g., location, organization) and shows its limits with domain-specific and long-tail entity types. Consider the following sentence: "We evaluated the performance of SimFusion+ on the WebKB dataset". Despite WebKB being a popular dataset in the Web research community, generic NER (e.g., Textrazor) can identify it as an entity but mistype it as an *Organization* instead of the domain specific entity type *Dataset*. We hypothesize that by using the existing pre-trained NER, we can identify the entities mentioned in the text. However, since existing NER is not trained for long-tail entity types and is not able to assign a label to the extracted entities, we first classify the sentences in a given text into predefined entity types using distant supervision. Next, we use existing pre-trained NER to extract the long-tail entities from the classified sentences and assign them the type matching the sentence class. Chapter 2 addresses RQ 1 and focuses on long-tail entities related to *Dataset*, *Method*, *Software*, *Objective*, and *Result*. The content of this chapter is based on the following publication:

  – Sepideh Mesbah, Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Semantic annotation of data processing pipelines in scientific publications. In: Extended Semantic Web Conference (ESWC), pp. 321-336, Springer, 2017.

  The results show that we can extract (rather noisy) entities with minimal human supervision, which we subsequently filter and rank, to select entities that promise high descriptive power for their class. While promising, generic NER shows its limits with domain-specific and long-tail entity types. The results suggested that as further improvement, there is a need to train domain-specific NER. For this, we require training data for a given entity type, which is hard to obtain. This leads us to our next research question:

- **RQ 2:** How can semantic expansion techniques and filtering heuristics be leveraged to augment training data for L-tER?

We hypothesize that there are recurring patterns in the mentions of long-tail entities and that they appear in similar contexts. With this hypothesis in mind, we designed TSE-NER, an iterative approach for long-tail entity extraction. TSE-NER uses semantic expansion strategies together with heuristic filters, which rely on minimal human input, a seed set of instances of the targeted entity type. Chapter 3 addresses RQ 2 and focuses on long-tail entities related to entities types *Dataset*, *Method* in computer science publications, and *Proteins* in biomedical publications. The content of this chapter is based on the following publication:

  - Sepideh Mesbah, Christoph Lofi, Manuel Valle Torre, Alessandro Bozzon, and Geert-Jan Houben. TSE-NER: An iterative approach for long-tail entity extraction in scientific publications. In International Semantic Web Conference (ISWC), pp. 127-143, Springer, 2018.

The results show that we can tune the technique for either higher recall (up to 0.41) or higher precision (up to 0.91) scenarios with only a small set of seed names (i.e., 5 - 100). While promising, we see that the precision drops after several iterations due to the simple heuristic filtering. As these heuristics are prone to failure, the overall achievable performance is limited. This leads us to our next research question, where we try to incrementally incorporate human feedback on the relevance of extracted entities into the training cycle of such iterative TSE-NER algorithms to improve the overall performance concerning precision, recall, and F-measures.

- **RQ 3**: How can collaborative feedback from human annotators be leveraged to improve L-tER?

  We hypothesize that by incorporating user feedback into the TSE-NER training process, we can augment the filtering step of TSE-NER to improve the overall performance. The human-in-the-loop approach allows us to maintain the advantages of the initial design of TSE-NER (i.e., training a NER algorithm cheaply, only relying on a small seed set, and providing an immediate result to users with acceptable extraction quality) while exploiting the human feedback into the next TSE-NER training iteration. For this, we introduce Coner, an approach that allows the users of our system to continuously provide easy-to-elicit low-effort feedback on the semantic fit and relevance of extracted entities. Chapter 4 addresses RQ 3 and focuses on long-tail entities related to entities types *Dataset*, *Method* in computer science publications. The content of this chapter is based on the following publication:

  - Daniel Vliegenhart, Sepideh Mesbah, Christoph Lofi, Akiko Aizawa, Alessandro Bozzon. Coner: A Collaborative Approach for Long-tail Named Entity Recognition in Scientific Publications. In International Conferences on Theory and Practice of Digital Libraries (TPDL), pp. 3-17, Springer, 2019.

Our experiments show that with *Coner*, we can decrease the number of false positives and false negatives. Furthermore, we show that by obtaining feedback on only 0.05% of the entities in the test set (and others outside the set), we could increase the precision by 4% while keeping recall and f-score stable. However, the experiments were conducted in a private lab experiment with only 15 graduate-level/post-graduate-level volunteers. For future work, we can leverage Coner's full potential by integrating it into an existing production system, like a large scale digital library. In this case, we can receive continuous feedback from the system's users on several papers, magnitudes bigger than our private lab experiment conducted so far and improve the performance of the TSE-NER models over time.

While the techniques introduced in Chapters 2, 3 and 4 have indeed shown to reduce the cost of training and improve the overall performance of Long-tail Entity Recognizer, they are typically limited by the availability of the words and sentences in the semantic space (Chapter 3) and the availability of continuous feedback from users (Chapter 4). This leads us to our next research question where we focus on generating new text not existing in the corpus, thus largely expanding the training data in a cost efficient manner:

- **RQ 4:** How can deep generative models be leveraged to improve the performance of L-tER?

  We hypothesize that by leveraging deep probabilistic modeling to capture the underlying data structure, we can automatically generate large training datasets from a small number of labeled samples. For realizing this goal, we modified Variational Autoencoders [21] in such a way that we can generate new realistic artificial training sentences from a given corpus resembling the subset of the corpus for which human annotation is available. Then, we heuristically annotate the new sentences by propagating the labels. As another example, user-generated phrases are examples of long-tail entities present in User Generated Content (UGC) published in online communication platforms such as Twitter or Reddit Chapter 5 addresses RQ 4. Until now Scientific publications were the main datasource used in our research to augment training data for the extraction and typing of long-tail entities. To further our understanding of how to augment training data for the extraction and typing of long-tail entities in other sources, we look into User generated content (UGC). UGC such as Twitter messages is noisy text often containing ungrammatical sentence structures and non-standard words in contrast to the text of Scientific publications which is structured. In Chapter 5 we focused on long-tail entities related to entity type *Adverse Drug Reaction* (ADR) in UGC, such as Twitter and Reddit. The content of this chapter is based on the following publication:

  - Sepideh Mesbah, Jie Yang, Robert-Jan Sips, Manuel Valle Torre, Christoph Lofi, Alessandro Bozzon, Geert-Jan Houben. Training Data Augmentation for Detecting Adverse Drug Reactions in User-Generated Content. In: Inter-

national Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 2349-2359, 2019.

An extensive evaluation performed on Twitter and Reddit data shows that our approach has comparable performance to fully supervised techniques while drastically lowering the demand for labeled training data, allowing us to maintain performance with down to only 25% of training data. However, there is a saturation effect: when sufficient manual training data is available, further artificial data generation has only limited positive effects. This limitation is likely due to our constraint to generate sentences similar to the existing annotated sentences instead of radically new ones - a choice that allows us to perform reliable label propagation, which would be hard for sentences that are too different.

1.3 Original Contribution

In this thesis we make the following contributions:

- In Chapters 2-5, we focus on RQ 1 - RQ 4, and we contribute novel techniques for augmenting training data to support the supervised training of L-tER with low training and re-training costs. The code is available at the following address: `https://github.com/mesbahs/TSE-NER`
- To evaluate our approach, we contribute two annotated datasets for the extraction and typing of long-tail entities in Scientific publications (in Chapter 3) and User Generated Content (Chapter 5). The dataset is available at the following address: `https://github.com/mesbahs/ADR_EMNLP`
- We contribute a novel web-based platform that supports the exploration and visualization of long-tail entities in scientific Publications (i.e., the architecture and functionalities are presented in Appendix A). A demo version of the platform is available at the following address: `https://smartpub.tk`.

In addition to the contributions mentioned above, during my doctoral studies, I focused on related research projects in the area of *Information Extraction*, which were published as peer-reviewed papers (i.e., one is still under review). We investigated Multilingual Open Relation Extraction (ORE) when limited training data is available (Appendix B). We further looked into Normalizing Adverse Drug Reactions (ADR) reports from user-generated content to concepts in a controlled medical vocabulary (Appendix C). We designed an ontology to support the description and encoding of relevant properties of long-tail entities found in scientific publications (Appendix E). Finally, we looked at the applications of the extracted long-tail entities in the digital library domain (Appendix D) and in real-life MOOCs (Appendix F).

1.4 Publication List

To provide an overall perspective of the research work carried out during my doctoral studies, a complete list of publications is presented below:

- Sepideh Mesbah, Jie Yang, Robert-Jan Sips, Manuel Valle Torre, Christoph Lofi, Alessandro Bozzon, Geert-Jan Houben. Training Data Augmentation for Detecting Adverse Drug Reactions in User-Generated Content. In International Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2349-2359, 2019. [This thesis]
- Daniel Vliegenhart, Sepideh Mesbah, Christoph Lofi, Akiko Aizawa, Alessandro Bozzon. Coner: A Collaborative Approach for Long-tail Named Entity Recognition in Scientic Publications. In International Conferences on Theory and Practice of Digital Libraries (TPDL), pp. 3-17, Springer, 2019. [This thesis]
- Sepideh Mesbah, Christoph Lofi, Manuel Valle Torre, Alessandro Bozzon, and Geert-Jan Houben. TSE-NER: An iterative approach for long-tail entity extraction in scientific publications. In International Semantic Web Conference (ISWC), pp. 127-143, Springer, 2018. [This thesis]
- Sepideh Mesbah, Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Semantic annotation of data processing pipelines in scientific publications. In Extended Semantic Web Conference (ESWC), pp. 321-336, Springer, 2017. [This thesis]
- Sepideh Mesbah, Alessandro Bozzon, Christoph Lofi, and Geert-Jan Houben. Smart-Pub: a platform for long-tail entity extraction from scientific publications. In Companion Proceedings of The Web Conference (TWC), pp. 191-194. 2018 (Appendix A).
- Tom Harting, Sepideh Mesbah, Christoph Lofi. LOREM: Language-consistent Open Relation Extraction from Unstructured Text, In The Web Conference (TWC), 2020 (Appendix B).
- Emmanouil Manousogiannis, Sepideh Mesbah, Selene Baez, Zoltán Szlávik, Alessandro Bozzon, and Robert Jan Sips. A shot in the dark: Few-Shot Learning to Normalize long-tail Adverse Drug Reaction Mentions on Twitter. In Journal of the American Medical Informatics Association (JAMIA) Journal (under review), 2020.
- Emmanouil Manousogiannis, Sepideh Mesbah, Alessandro Bozzon, Selene Baez, and Robert Jan Sips. Give it a shot: Few-shot learning to normalize ADR mentions in Social Media posts. In Proceedings of the Fourth Social Media Mining for Health Applications Workshop and Shared Task (SMM4H), pp. 114-116, 2019 (Appendix C).
- Sepideh Mesbah, Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Facet embeddings for explorative analytics in digital libraries.

In International Conference on Theory and Practice of Digital Libraries (TPDL), pp. 86-99. Springer, Cham, 2017 (Appendix D).

- Sepideh Mesbah, Alessandro Bozzon, Christoph Lofi, and Geert-Jan Houben. Describing data processing pipelines in scientific publications for big data injection. In Proceedings of the 1st Workshop on Scholarly Web Mining (SWM), pp. 1-8. 2017 (Appendix E).
- Sepideh Mesbah, Guanliang Chen, Manuel Valle Torre, Alessandro Bozzon, Christoph Lofi, and Geert-Jan Houben. Concept focus: semantic meta-data for describing MOOC content. In European Conference on Technology Enhanced Learning (EC-TEL), pp. 467-481. Springer, Cham, 2018 (Appendix F).
- Sarah Bashirieh, Sepideh Mesbah, Judith Redi, Alessandro Bozzon, Zoltán Szlávik, and Robert-Jan Sips. Nudge Your Workforce: A Study on the Effectiveness of Task Notification Strategies in Enterprise Mobile Crowdsourcing. In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP), pp. 4-12. 2017.

Chapter 2

Using Pre-trained NER for Recognizing Long-tail Entities

In this chapter we address RQ1 by investigating to what extent pre-trained generic NER can be used to recognize the long-tail entities. The downside of the generic NER is that it is not able to assign a type to the long-tail entities. To overcome this problem, we first describe a method designed to classify sentences of the scientific text containing domain-specific entities according to the nature of the contained information. Next, we use the existing NER to extract relevant named entities from the classified sentences. In this chapter we focus on domain-specific entity types such as scientific Objective, Dataset, Method, Software and Result, which are a core object of interest for data scientists and practitioners operating in a variety of data-related application domains. The extracted information is then semantically annotated. To demonstrate the effectiveness and performance of our approach, we present the results of a quantitative and qualitative analysis performed on four different conference series. The contribution of this chapter is published in [143].

2.1 Introduction

In scientific publications, scientists and practitioners *share* and *seek* information about the properties and limitations of 1) data sources; and 2) of data processing methods (e.g. algorithms) and their implementations. For instance, a researcher in the field of urban planning could be interested in *discovering state of the art methods for point of interest recommendation (e.g. matrix factorisation) that have been applied to geo-located social media data (e.g. Twitter) with good accuracy results.*

A system able to answer the query above requires access to a structured representation of the knowledge contained in one or more scientific publication repositories. For instance, it should be possible to *access* and *relate* information about: 1) the objective of a given scientific work; 2) the datasets employed in the work; 3) the methods (i.e. algorithms) and tools (e.g. software) developed or used to process such datasets; and 4) the obtained results.

Our vision is to offer support for semantically rich queries focusing on different aspects of data processing pipelines (e.g. methods, datasets, goals). The availability of a semantically rich, interlinked, and machine readable descriptions (metadata) of such knowledge could provide great benefits in terms of retrieval quality, but also for analysing and understanding trends and developments.

Manually inspecting and annotating papers for metadata creation is a non-trivial and time-consuming activity that clearly does not scale with the increasing amount of published work. Alas, scientific publications are also difficult to process in an automated fashion. They are characterised by structural, linguistic, and semantic features that are different from non-scientific publications (e.g. blogs). In this context, general-purpose text mining and semantic annotation techniques might not be suitable analysis and tools. As a consequence, there is a clear need for methodologies and tools for the extraction and semantic representation of scientific knowledge. Recent work focused on methods devoted to the automatic creation of semantic annotations for text snippets, with respect to either structural [104, 25, 185], argumentative [120, 76], or functional [128, 206, 165] components of a scientific work. However, to the best of our knowledge, there has been no work yet focusing on extracting metadata focusing on properties of data processing pipelines. Therefore, in this thesis, we provide the following contributions:

- A novel approach for the classification of text related to data processing pipelines from scientific publications, and for the extraction of named entities. The approach combines distant supervision learning on rhetorical mentions with named entity recognition and disambiguation.

Our system automatically classifies sentences and named entities into five categories (objectives, datasets, methods, software, results). Sentence classification attains an average accuracy of 0.80 and average F-score 0.59.

- A quantitative and qualitative evaluation of the implementation of our approach, performed on a corpus of 3,926 papers published in 4 different conference series

in the domain of Semantic Web (ESWC), Social Media Analytics (ICWSM), Web (WWW), and Databases (VLDB).

We provide evidence of the amount and quality of information on data processing pipelines that could be extracted, and we show examples of information needs that can now be satisfied thanks to the availability of a richer semantic annotation of publications' text. The remainder of the paper is organised as follows: Section 2.3 introduces the DMS ontology; Section 2.4 describes the data processing pipelines knowledge extraction workflow; Section 2.5 reports the results of the evaluations; Section 2.2 describes related work. Finally, Section 2.6 presents our conclusions.

2.2 Related Work

In the last few years there has been a growing interest in the open and linked publication of metadata related to scientific publications. There are now several ontologies devoted to the description of scholarly information (e.g. SWRC,[1] BIBO,[2] DMS [137]). The Semantic Dog Food [158] and the RKBExplorer [68] are examples of projects devoted to the publication of "shallow" meta data about conferences, papers, presentations, people, and research areas. A large portion of such shallow metadata is already explicitly given by the authors as part of the final document, such as references, author names, keywords, etc. Still, the extraction of that metadata from a layouted document is complex, requiring specialized methods [124] being able to cope with the large variety of layouts or styles used in scientific publication. In contrast, "deep" metadata as for example the topic, objectives, or results of a research publication pose a greater challenge as such information is encoded in the text itself. The manual creation of such metadata related to scientific publications is a tedious and time-consuming activity. Semi-automatic or automatic metadata extraction techniques are viable solutions that enable the creation of large-scale and up-to-date metadata repositories. Common approaches focus on the extraction of relevant entities from the text of publications by means of ruled-based [185, 76], machine learning [104], or hybrid (combination of rule based and machine learning) [206, 165] techniques.

These approaches share a common assumption: as the number of publications dramatically increases, approaches that exclusively rely on dictionary-based pattern matching (possibly based on pre-existing knowledge bases) are of limited effectiveness. Rhetorical entities (REs) detection [87] is a class of solutions that aims at allowing the identification of relevant entities in scientific publications by analysing and categorising spans of text (e.g. sentences, sections) that contain information related to a given structural [104, 25, 185] (e.g. Abstract, Introduction, Contributions, etc.), argumentative [120, 76] (e.g. Background, Objective, Conclusion, Related Work and Future Work), or functional (e.g. datasets [128], algorithms [206], software [165]) classification.

[1] http://ontoware.org/swrc/

[2] http://bibliontology.com

In contrast to existing literature, our work focuses on rhetorical mentions that relate to the description (Objective), implementation (Dataset, Method, Software), and evaluation (Result) of data processing pipelines. Thanks to a distant supervision approach and a simple feature model (bags-of-words), our method does not require prior knowledge about relevant entities [128] or grammatical and part-of-speech characteristics of rhetorical entities [206]. In addition, while in previous work [25, 185] only one or few sections of the paper (e.g abstract, introduction) are the target of rhetorical sentences classification, we make no assumption about the location of relevant information. This adds additional classification noise, due to the uncontrolled context of training sentences: it is more likely for a "Result" section to describe experimental results than for a "Related Work" section, where the likelihood of misclassification is higher [87].

2.3 The DMS Ontology

The DMS (Dataset, Method, Software) ontology [137] is designed to support the description and encoding of relevant properties of data processing pipelines, while capitalising on established ontologies. DMS has been created in accordance to the *Methondology* guidelines [62]. It has been implemented using OWL 2 DL, and it consists of 10 classes and 30 properties. DMS captures five main concepts, namely *objectives*, *datasets*, *methods*, *software*, and *results*.

In the following, we refer to this initial ontology as DMSC. We provide an overview of the five aforementioned core concepts in Figure 2.1 (in order to keep compatibility with existing ontologies, for some concepts, we adopt slightly different naming conventions within the ontology and in this text, i.e., *dataset* is encoded as *disco:DataFile* in DMS). Data processing pipelines are composed of one or more methods (*deo:Methods*), and are typically designed and evaluated in the context of a scientific experiment (*dms:Experiment*) described in a publication (*dms:Publication*). An experiment applies data processing methods, implemented by software (*ontosoft:Software* [67]), to one or more datasets (*disco:DataFile*) in order to achieve a given objective (*dms:Objective*), yielding one or more results (*deo:Results*). In each experiment, different implementations or configurations of a method (*dms:MethodImplementation*) or software (*dms:softwareconfiguration*) can be used. However, in this work, we only focus on the core concepts ignoring configurations and implementations.

Our main contribution in this chapter is a methodology for the automatic extraction of metadata in accordance with the five core concepts of DMS: objective, dataset, method, software, and result. We reach this goal by labeling each of the sentences in a publication when it contains a *rhetorical mention* of one of the five DMS concepts. To capture knowledge on the properties and results of this extraction process, we introduce an auxiliary module DMSR (Figure 2.1) extending DMSC as discussed in the following. `DMS-rhetorical` allows to link any *dms:CorePipelineConcept* (i.e. the supertype of *objective*, *dataset*, *method*, *software*, and *result*) to an extracted rhetorical mention.

This link includes relevant provenance information such as the source of that mention (e.g. the sentence and section within a publication), but also metadata related

to the extraction process, such as the classifier used to associate a sentence to a given DMS concept, and the related classification confidence.

We reuse the DoCo [39] ontology for encoding the information on sections and sentences. For each publication, we keep its general metadata including *id*, *title*, *authors*, *year of publication*, and *publisher*. The publication contains (*pattern:contains*) sections and each section of the paper contains several sentences. We store the text of the sentence using the *doco:Sentence* class and link the sentence *pattern:contains* to its *dms:CorePipelineConcept*.

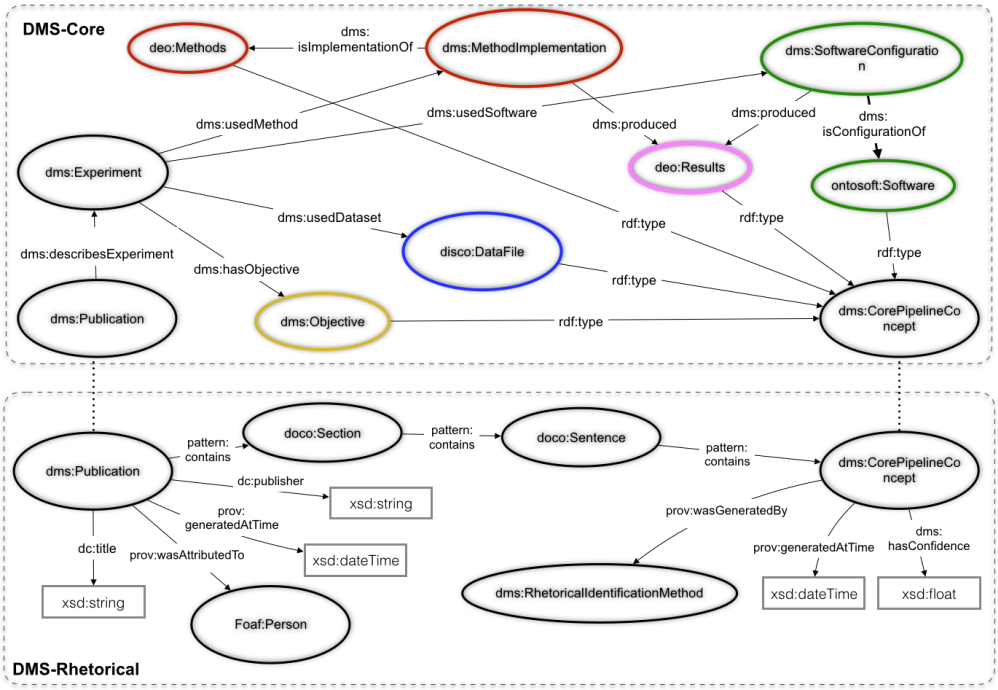


Figure 2.1: DMSC ontology and the DMSR extension.

2.4 DPP Knowledge Extraction Workflow

This section presents the knowledge extraction workflow designed to identify and annotate information referring to data processing pipelines (DPP) along the lines of the main classes of the DMS ontology (i.e. datasets, methods, software, results, and objectives). Our whole approach is summarized in Figure 2.2. First, we identify rhetorical mentions of a DMS main class. In this work, for the sake of simplicity, rhetorical mentions are sought at sentence level. Future works will introduce dynamic boundaries, to capture the exact extent of a mention. Then, we extract named entities from the rhetorical mentions.

These entities are filtered and, when applicable, linked to pre-existing knowledge bases, creating the final knowledge repository.

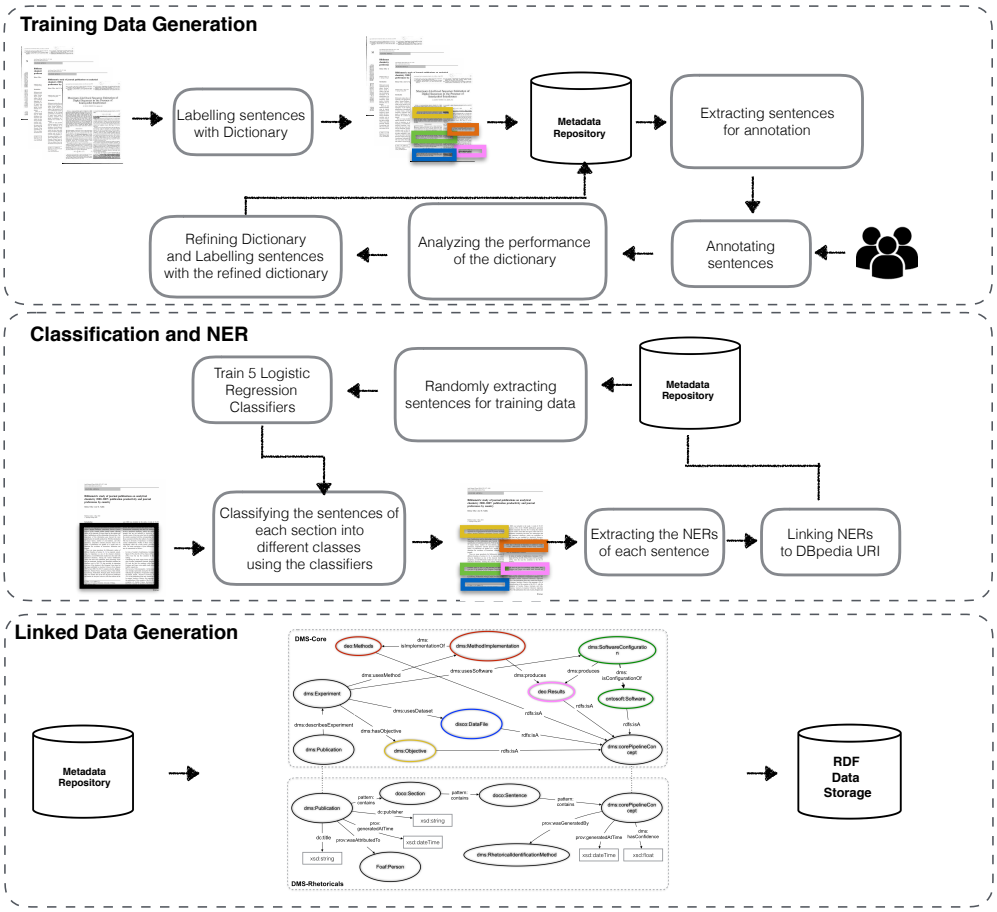


Figure 2.2: Data Processing Pipeline Knowledge extraction workflow.

The identification of rhetorical mentions is obtained through a workflow inspired by distant supervision [154], a training methodology for machine learning algorithms that relies on very large, but noisy, training sets. The training sets are generated by means of a simpler classifier, which could rely, for instance, on a mix of expert-provided dictionaries and rules, refined with manual annotations. Intuitively, the training noisiness could be cancelled out by the huge size of the semi-manually generated training data. This method requires significantly less manual effort, while at the same time retaining the performance of supervised classifiers. Furthermore, this approach is more easily adapted to different application domains and changing language norms and conventions.

Training Data Generation

Data Preparation

Scientific publications, typically available in PDF, are processed using one of the best-state-of-art extraction engines, GeneRation Of BIbliographic Data (GROBID) [130, 124]. GROBID extracts a structured full-text representation as Text Encoding Initiative(TEI)-encoded documents, thus providing easy and reliable access paragraphs and sentences.

Dictionary-based Sentence Annotation

Our goal is to classify each sentence of a given publication with respect to the five main classes of the DMS Ontology (datasets, methods, software, results, and objectives), based on the presence of rhetorical mentions that are related to such classes. Sentence classification could be obtained by means of a traditional supervised machine learning approach, assuming the presence of a large enough training set of sentence-level annotations. In our previous work E, we manually created a small set of high-quality sentence-level annotations, relying on expert feedback. However, the annotation of a single publication took around 30-60 minutes per annotator, showing that this approach was not sufficiently scalable. We therefore opted for a workflow inspired by *distant supervision*. All sentences in our corpus were automatically labeled using a lower-quality and noisy dictionary-based classifier and simple heuristic rules, which are created using the following two-steps approach:

- **Reuse of generic scientific rhetorical phrases**: We relied on manually curated and published dictionaries of phrases and words found in [53] and [1] as an initial starting point to build our own dictionary. Both papers are writing guides giving advise on how to write an academic text based on best practices and commonly used phrases. [1] covers common phrases for introducing different sections in academic literature, e.g. the abstract, problem statement, methodology, or result discussion. [53] presents an extensive manual corpus study on different parts of scientific argumentation, and gives suggestion for accepted and often used phrases split by different disciplines and publication types.
- **Manual refinement and adaptation to the DMS domain**: The set of dictionary words based on [53] and [1] did not focus specifically on rhetorical mentions of data processing pipelines (even though classes like "result discussion" are quite related). Therefore, we manually refined those dictionaries and adapted them specifically to our 5 DMS classes. This refinement is based on the careful inspection of 20 papers selected from four Web- and data- related conferences series (ESWC, VLDB, ICWSM, and WWW).

The outcome of these two steps is a more class-specific set of dictionaries. For example the rhetorical phrases *"we collected"* and *"we crawled"* indicate a rhetorical mention of the *dataset* class. We used the dictionary to label sentences of 10 publications randomly selected from the four conferences series, to manually check the performance of

the dictionary. For instance, we observed that the word *"data"* alone in a sentence is not a good indicator for being related to *dataset*. However if the word *"data"* co-occurs with *"from"*, a relationship with *dataset* is more likely. Several iterations of this manual refinement process lead to the final dictionary used for the following steps. Some example phrases are shown in Table 2.1.[3] Note that rhetorical mentions used in our refined dictionary are in fact skip n-grams, i.e. we do not expect the terms of each skip n-gram to be adjacent in a sentence (e.g. the rhetorical mention "the aim of this study" stripped of stop words becomes the skip n-gram "aim study").

Objective	*this research, this article, aim study, aim article, purpose paper, we aim, we investigate*
Dataset	*dataset, datasource, data source, collected from, database, collect data, retrieve data*
Method	*we present, we develop, we conduct, we propose, methodologies, method, technique*
Sofware	*tool, obtained using, collected using, extracted using, software*
Result	*we find, shows, show, shown, showed, we found, figure, table, we observe, we compare*

Table 2.1: Excerpt of dictionary of phrases used for classifying sentences

Test and Training Data Generation

We created reliable test and training datasets for both training and benchmarking machine learning classifier as follows. By using the phrases dictionary described in the previous subsection, we label all sentences of all research papers collected with appropriate class labels. Most sentences will not receive a label (as they do not contain any rhetorical mentions), but some may obtain multiple labels. This is for instance common for sentences found in an abstract, which often contain information on *dataset*s, but also on *method*s, or even *result*s. Then, we randomly select a balanced set of sentences with rhetorical mentions of all five classes, and manually inspect the assigned labels. We reclassify them using expert feedback from several annotators, if the pattern-based classifier assigned incorrect labels. Using this approach, we can create a reliable manually annotated and balanced test dataset quicker and cheaper compared to annotating whole publications or random sentences, as the pattern-classifier usually delivers good candidate sentences. Furthermore, this approach allows us to further refine and improve the dictionary by incorporating the expert feedback, allowing us to cheaply re-annotate the whole corpus using the dictionary with higher accuracy compared to the initial classifier.

We assessed the performance of both the dictionary-based classifier and our annotators to decide on the number of manual annotations needed for a reliable test set. We randomly selected 100 sentences from each of the five classes (i.e. 500 in total). Two expert annotators manually checked the assigned labels (a task which was perceived easier

[3]The dictionaries are available at `https://github.com/WISDelft/SmartPub/blob/master/playground/dictionary.py`

by the annotators than applying labels to a random unlabeled sentence). The inter-annotator agreement using the Cohen's kappa measure averaged over all classes was .58 (the Cohen's kappa measures of the individual classes are *objective*: .71, *dataset*: .68, *software*: .37, *result*: .61, and *method*: .53).

Classification and NER

Machine-Learning-based Rhetorical Detection

As a second part of our distant supervision workflow, we now train a simple binary Logistic regression classifier for each of the classes using simple TF-IDF features for each sentence. This simple implementation serves as a proof of concept of our overall approach, and can of course be replaced by more sophisticated features and classifiers in future work.

As a test set, we use the 500 sentences (100 per class) manually labeled with their DMS class by our expert annotators. We associated a single label (some sentences can have multiple labels) to each sentence, decided by a simple majority vote. In order to generate the training data for each class, we randomly selected 5000 positive examples from the sentences labeled with that class by the dictionary-based classifier. We also randomly select 5000 negative examples from sentences which are not labeled with that class by the dictionary classifiers. Sentences from the test set were excluded from the pool of candidate training sentences.

Named Entity Extraction, Linking, and Filtering

In the last step of our method, we extract named entities from the sentences that are classified as related to one of the five main DMS classes, filtering out those entities that are most likely not referring to one of the DMS classes, and retaining the others as an extracted entity of the class matching the sentence label.

Named entity extraction has been performed using the TextRazor API[4]. TextRazor returns the detected entities, possibly decorated with links to the DBpedia or Freebase knowledge bases. As we get all named entities of a sentence, the result list contains many entities which are not specifically related to any of the five classes (e.g. entities like "software", "database"). To filter many of these entities, and after a manual inspection, we opted for a simple filtering heuristic. Named entities are assumed to be not relevant if they come from "common" English language (like software, database), while relevant entities are terms referring to domain-specific terms or specific acronyms (like SVM, GROBID, DMS, Twitter data). The heuristic is implemented as look-up function of each term in *Wordnet*.[5] Named entities that can be found in WordNet are removed. As WordNet is focusing on general English language, only domain-specific terms remain. We present the results of the analysis performed on the quality of the remaining named entities in Section 2.5.

[4] http://www.textrazor.com/

[5] http://wordnet.princeton.edu/

Linked Data Generation

As a final step, we build a knowledge repository based on the DMS-Core and DMS-Rhetorical ontology (outlined in Section 2.3). The repository is populated with classified sentences, and with the lists of entities for each DMS main class, with links to the sentence where each single entity has been detected. Sentences are linked to the containing publications.

Listing 2.1 shows an example of a part of an output RDF. The relationships shown in the RDF snippet are from the domain-specific DMS ontology for describing data-processing research. They have not been extracted automatically, as the scope of this work is not on the automatic extraction of relationships between entities.

```
PREFIX doco: <http://purl.org/spar/doco>
PREFIX prov: <http://www.w3.org/ns/prov>
PREFIX disco: <http://rdf-vocabulary.ddialliance.org/discovery>
PREFIX dms: <https://github.com/mesbahs/DMS/blob/master/dms.owl>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns>
PREFIX pattern: <http://www.essepuntato.it/2008/12/pattern>
[a dms:Publication;
dms:describesExperiment  dms:Ncdec5e68ed864a3a24].
dms:Ncdec5e68ed864a3a24   a  dms:Experiment;
      dms:usedDataset [ a   disco:dataFile ;
                        rdf:type      dms:Ncdec5e68ed864a ;
                        prov:value     "Billion Triple Challenge (BTC)"].
dms:Ncdec5e68ed864a   a   dms:CorePipelineConcept;
       pattern:isContainedBy  doco:Ncdec5e68edghgf99.
doco:Ncdec5e68edghgf99 a doco:Sentence;
       prov:value "In our experiments we used real data that were taken from the Billion Triple Challenge (BTC) dataset.";
       pattern:isContainedBy doco:Ncdec5ehfdjk67.
doco:Ncdec5ehfdjk67 a doco:Section;
       prov:value "Introduction".
```

Listing 2.1: Example of output RDF: A paper describes an experiment which uses a dataset called (BTC). (BTC) is a CorePipelineConcept linked to sentence of the paper.

2.5 Evaluation

In this section, we analyse the performance of our metadata extraction pipeline in both a quantitative and qualitative fashion. We focused on four major conference series from different communities with notable scientific contributions to data processing pipelines (Table 2.2): the European Semantic Web Conference (ESWC), International Conference On Web and Social Media (ICWSM), International Conference on Very Large Databases (VLDB), and the International World Wide Web Conference (WWW). We further present the results of both the dictionary-based and logistic regression-based sentence classifiers on the manually annotated test data. Finally, we analyse and discuss the quality of the entities extracted from the classified sentences.

Dataset

Table 2.2 summarises the properties of the experimental dataset, including its size, the number of rhetorical mentions extracted for each class (as decided by the regression-based classifier), and the number of unfiltered unique named entities extracted from the rhetorical mentions taken from scientific publications of a particular conference series. The table shows that methods are the most frequent encountered class, followed by datasets. Table 2.3 summarises statistics on extracted entities as described in the previous section per class (including filtering and pruning entities using a Wordnet look-up). Furthermore, we report how many of those entities could be linked to Wikipedia by the TextRazor API (columns *with URI*), thus distinguishing well-known entities (e.g. Facebook, Greedy algorithm) from the newly presented or less popular entities (e.g. SIFT Netnews, RW ModMax. columns *no URI*).

Conf.	Size		Rhetorical sentences					Unique Named Entities				
	#PAP	#SNT	#OBJ	#DST	#MET	#SWT	#RES	#OBJ	#DST	#MET	#SWT	#RES
ESWC	620	129760	12725	13528	26337	9614	22245	4197	4910	6987	4557	6416
ICWSM	793	52094	6096	4277	8936	1830	13848	2830	2241	3658	1538	4499
VLDB	1492	396457	26953	49855	68336	11919	84662	7301	12052	13920	5741	15959
WWW	1021	253401	23378	19783	49331	10293	58212	6616	6499	10793	5164	11869

Table 2.2: Quantitative analysis of the rhetorical sentences and named entities extracted from four conference series. Legend: PAP (papers), SNT (sentences), OBJ (objective), DST (dataset), MET (method), SWT (software), RES (results)

Conf.	Distinct NER with URI					Distinct NER no URI				
	#OBJ	#DST	#MET	#SWT	#RES	#OBJ	#DST	#MET	#SWT	#RES
ESWC	1157	1206	1779	1200	1454	1874	2427	3497	2193	3219
ICWSM	727	555	944	443	1027	1110	900	1588	519	1974
VLDB	1528	2313	2516	1365	2395	3800	6963	8393	2804	10288
WWW	1990	1630	2904	1613	2860	2742	3153	5382	2148	6247

Table 2.3: Number of Named Entities after filtering using the Wordnet.

ESWC	***ICWSM***	***VLDB***	***WWW***
Semantic Web	LDA	Tuple	Web Page
Sem-CF	Classifier_I	XML	Login
User Modeling	SetLock	Query Plan	Faceted Search
Recommender System	Hashtag	XsKetch	Recommender System
FactBox	Future tense	LS-B	Source Rank

Table 2.4: Top-5 most frequent methods applied to IMDB dataset.

Qualitative Analysis

In this section, we showcase how our approach can be used to fulfill a hypothetical information need of a data scientist, namely: *Which methods are commonly applied to a given data set?*

As an example, we use the popular IMDB dataset of movies and actors, and manually inspect the list of top-6 most frequent methods applied to that dataset in publications grouped by their conference series. The results are shown in Table 2.4, hinting at the different interests conference venues have for that dataset: ignoring the false positives (like "Web Page" or "XML" - we further discuss false positives later in this section), VLDB as a database-centric conference covers methods like XsKetch (summarisers for improving query plans in XML databases) or LSB-Trees for better query plans for nearest-neighbour queries, using the IMDB dataset as a large real-life dataset for evaluation database queries; ICWSM with a focus on Social Media research features LDA topic detection and generic classification to analyse IMDB reviews, while ESWC and WWW are interested in recommendations and user modelling.

Analysis of Rhetorical Classifiers

In the following, we present the results of both the *dictionary-based* and *logistic regression-based* classifiers on the manually annotated test set, summarised in Table 2.5, relying on commonly used measurements for accuracy, precision, recall, and F-Score. It can be observed that using logistic regression increases the recall for most classes, while having a slightly negative impact on the precision, showing that this approach can indeed generalise from the manually provided dictionaries to a certain extent.

We believe that better performance can be achieved by employing more sophisticated features and classifiers. Furthermore, the performance gains of the logistic regression classifier come for "free" as we only invested time and effort to train the dictionary-based classifier. The best results are achieved for the *Method* class with F-score=0.71. We manually inspected the sentences labeled as *Software* and *Dataset* to understand reasons for the comparatively low performance of those classes. To certain extend, this can be attributed to the ambiguity of some n-grams in the dictionary. For example, the word *tool* appearing in different sentences can result to misleading labels: e.g., "extraction tool Poka" is about software, but "current end-user tools" is a general sentence not specifically about a software. Similarly confusion can be observed for the word *dataset* for the *Dataset* class. For instance, "twitter dataset" and "using a dataset of about 2.3 million images from Flickr" are labeled correctly, but "quadruple q and a dataset d" is labeled incorrectly. Thus, we conclude that many terms used in *Software* and *Dataset* are too generic (e.g. dataset, tool, database) leading to higher recall, but having a negative impact on precision, demanding more refined rules in our future work.

	Dictionary based				Logistic regression based			
Classes	*Accuracy*	*Precision*	*Recall*	*F-Score*	*Accuracy*	*Precision*	*Recall*	*F-Score*
Objective	0.85	0.49	0.81	0.61	0.84	0.49	0.81	0.61
Dataset	0.84	0.46	0.68	0.55	0.80	0.41	0.81	0.54
Method	0.76	0.79	0.61	0.69	0.76	0.76	0.67	0.71
Software	0.83	0.39	0.52	0.45	0.84	0.34	0.72	0.46
Result	0.84	0.60	0.68	0.63	0.81	0.53	0.71	0.60

Table 2.5: Estimated Accuracy, Precision, Recall and F-score on manually annotated sentences for Dictionary and Logistic Regression based classification

Quality of Extracted Entities

We studied the performance of the Named Entity (NE) extraction modules of our method by means of a mixed quantitative and qualitative analysis. We calculated the Inverse Document Frequency (IDF) of each named entity NE_i extracted from the corpus. IDF is a measure of informativeness, calculated as $IDF(NE_i) = log\frac{|Sentences|}{|NE_i|}$, that is, the logarithmically scaled inverse fraction of the number of sentences in the corpus and the number of sentences containing NE_i. Figure 2.3 depicts the distribution of IDF values for each NE in the dataset.

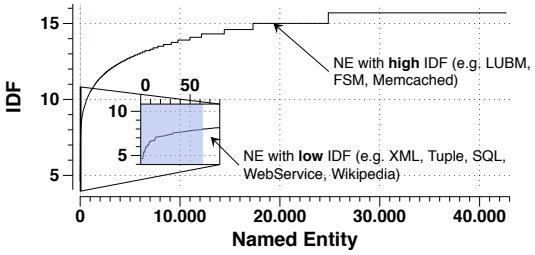


Figure 2.3: Distribution of IDF values of extracted named entities.

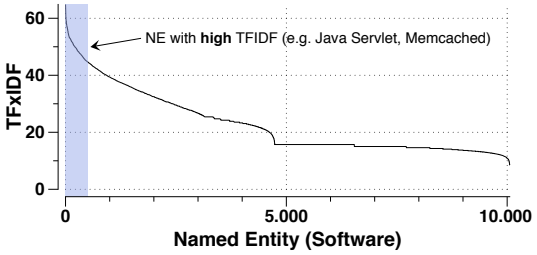


Figure 2.4: Distribution of TFIDF values for NEs contained in *software* sentences.

Only a handful of named entities (about 100) feature a low IDF values (indicating that they are likely not fitting their assigned class well), while a large amount of entities (more than 60%) have relatively high informativeness. But, what is the quality of such entities? Are they useful in the characterization of class-specific sentences? To answer these questions, we first calculated a class-specific TFxIDF value for each named entity NE_i in the dataset as $TFIDF(NE_i, C_j) = (1 + log(|NE_{i,j}|)) \times IDF_{NE_i}$, where $|NE_{i,j}|$ is the raw frequency of a named entity NE_i within the sentences classified as relate to the class C_j. Then, for each class, we ranked named entities in decreasing order of $TFIDF(NE_i, C_j)$, and manually analyzed the first 100 entities.

Figure 2.4 shows an example distribution of TFIDF values. We excluded from this analysis the *objective* class, as objectives are usually not represented well by a single named entity, but instead require a more elaborate verbal description (which is usually fittingly provided by a rhetorical mention).

Table 2.6 shows examples of relevant named entities for each considered class. In terms of retrieval precision, we can observe promising results. NEs contained in *method* and *software* sentences feature a precision of 72% and 64%, respectively. On the other hand, NEs contained in *dataset* and *results* sentences resulted in a precision of 23% and 22%. In both cases, however, the returned entities are still relevant and related to the class: False positives in *dataset* sentences are mainly due to terms that are clearly related to data (e.g. Fuzzy set, Data model, Relational Algebra), but not specifically referring to actual datasets. Likewise, false positives in *results* sentences are mainly due to the presence of acronyms that could be linked to the names of the methods tested in the paper. This type of error can be attributed the the sentence-level granularity of our rhetorical mention detection, and can likely be reduced by including a boundary classifier into our workflow.

In summary, we can conclude that our approach is indeed suitable for extracting entities with respect to the five DMS classes in a meaningful and descriptive fashion. However, there are still some false positives of related concepts which cannot easily be recognized using simple statistic means, and which thus invite further deeper semantic filtering in future works.

Dataset	***Method***	***Software***	***Result***
MovieLens	Collaborative Filtering	Java Servlet	Expected Value
Enron	Dynamic Programming	Portlet	Standard Deviation
IMDb	Active Learning	PHP	Precision and Recall
YAGO	Support Vector Machine	Memcached	P-value
DBPedia	Language Model	DOM API	MRR

Table 2.6: Examples of representative Named Entities in different classes

2.6 Conclusion

In this chapter, we focused on utilizing pre-trained NER to support the extraction and typing of long-tail entities. The workflow specializes on the extraction of domain-specific entities related to data processing pipelines, with a focus on rhetorical mentions related to Datasets, Methods, Software, Objectives, and Results. The extracted information is collected and published as a RDF knowledge base according to the DMS (Data Method Software) ontology, which was specifically designed to enable the description and linking of information related to data processing pipelines. The generated metatada allows researchers and practitioners to access and discover valuable information related to the properties and limitations of data sources and data processing pipelines, based on current literature.

Differently from previous work, our workflow relies on a lightweight distant supervision approach, which features lower training costs (compared to traditional supervised learning) and acceptable performance. These properties make the approach suitable for reuse in additional knowledge domains related to scientific publication. We show that,

despite its simple design, it is possible to achieve high precision and recall for all classes. From these classified sentences, we extracted (rather noisy) named entities, which we subsequently filtered and ranked, to select entities that promise high descriptive power for their class.

While promising, the obtained results suggest ample space for future improvements. For instance, it will be interesting to investigate the performance of more complex machine learning classifiers working on richer feature sets (e.g., word-embeddings, POS-tags, parse trees, etc.). Furthermore, for labeling scientific entities, our current granularity is on sentence level. This introduces some additional confusion when extracting named entities in cases that a sentence has multiple labels, or only parts of a sentence refer to a rhetorical mention while others do not. This limitation could be remedied by additionally training a Long-tail Entity Recognizer (L-tER) for a given entity type. For this, we require training data for a given entity type, which is hard to obtain. In Chapter 3 we tackle the problem of lack of training data using semantic expansion techniques.

Chapter 3

Training Data Augmentation by Exploiting Term and Sentence Expansion Strategies

L-tER is a challenging task, especially with entities such as the domain-specific ones found in scientific publications. These entities (e.g., "WebKB", "StatSnowball") are rare, often relevant only in specific knowledge domains, yet important for retrieval and exploration purposes. State-of-the-art NER approaches employ supervised machine learning models, trained on expensive type-labeled data laboriously produced by human annotators. A common workaround is the generation of labeled training data from knowledge bases; this approach is not suitable for long-tail entity types that are, by definition, scarcely represented in KBs. This chapter addresses RQ2 by presenting different strategies for training data augmentation to improve the supervised training of Long-tail Entity Recognition. Our technique starts with a minimal human input, namely a small seed set of instances for the targeted entity type and enhances the size of the training data using semantic expansion techniques automatically and iteratively. We evaluate our approach on scientific publications, focusing on the long-tail entities types *Datasets*, *Methods* in computer science publications, and *Proteins* in biomedical publications. The contribution of this chapter is published in [147].

3.1 Introduction

The growth of domain-specific knowledge available as digital text demands more effective methods for querying, accessing, and exploring document collections. Scientific publications are a compelling example: online digital libraries (e.g. IEEE Xplore) contain hundreds of thousands documents; yet, the available retrieval functionality is often limited to keyword/faceted search on *shallow* meta-data (e.g. title, terms in abstract). A query like *retrieve the publications that used a social media dataset for food recipe recommendation* is bound to return unsatisfactory results.[1]

Named entities, obtained through an analysis of a document's content, are an effective way to achieve better retrieval and exploration capabilities. Automatic *Named Entity* Recognition and Typing (NER/NET) is essential to unlock and mine the knowledge contained in digital libraries, as most smaller domains lack the resources for manual annotation work.

To perform well, state-of-the-art NER/NET methods [23, 101] either require comprehensive domain knowledge (e.g. to specify matching rules), or rely on a large amount of human-labeled training data for machine learning – both solutions are expensive and time-consuming.

A cheaper alternative is to generate labeled training data by obtaining existing instances of the targeted entity type from Knowledge Bases (KBs) [23] - this of course requires that the desired entity type is well-covered in the KB.

Problem Statement. While achieving impressive performance with high-recall named entities (e.g. locations and age) [101], generic NER/NETs show their limits with domain-specific and long-tail entity types. Consider the following sentence: *"We evaluated the performance of SimFusion+ on the WebKB dataset"*. Despite *WebKB*[2] being a popular dataset in the Web research community, generic NERs (e.g. Textrazor[3]) mistype it as an `Organization` instead of the domain-specific entity type `Dataset`. The entity *SimFusion+* of type `Software` is missed completely.

Literature [166, 185, 192, 200] shows that training of *domain-specific* NER/NETs is still an open challenge for two main reasons: 1) the *long-tail* nature of such entity types, both in existing knowledge bases *and* in the targeted document collections [177]; and 2) the high cost associated with the creation of hand-crafted rules, or human-labeled training datasets for supervised machine learning techniques. Few approaches addressed these problems by relying on bootstrapping [200] or Entity Expansion [23, 101] techniques, achieving promising performance. However, how to train high-performance *long-tail* Entity Extraction and Typing with minimal human supervision remains an open research question.

Original Contribution. We contribute TSE-NER, an iterative approach for training NER/NET classifiers for long-tail entity types that exploits Term and Sentence Expansion. TSE-NER relies on minimal human input – a seed set of instances of the targeted

[1] `https://scholar.google.de/scholar?q=publications+using++social+media+datasets+for+food+recipes+recommendation`

[2] `http://www.cs.cmu.edu/~WebKB/`

[3] `https://www.textrazor.com/`

entity type. We introduce different strategies for training data extraction, semantic expansion, and result entity filtering. Different combinations of these strategies allow to tune the technique for either higher recall or higher precision scenarios.

We performed extensive evaluations comparing to state-of-the-art methods, and assess several sentence expansion and term filtering strategies. As our core use case, we focus on 15,994 data science publications from 10 conference series with the *Dataset* (e.g. `Imagenet`) and data processing *Methods* (e.g. `LSTM`) long-tail entity types. We show that our approach is able to consistently outperform previous low-cost supervision methods, even with small amount of training information: with a seed set of 100 entities, our approach can achieve precision up to 0.91 when tuned for precision, and recall up to 0.41 when tuned for recall, or 0.77 and 0.30 for a balanced setting. When applied in an iterative fashion, our approach can achieve comparable performance with an initial seed set of only 5 entities. We show that sentence expansion and filtering strategies can provide a spectrum of performance profiles, suitable for different retrieval applications such as search (high precision) and exploration (high recall). To study the performance of TSE-NER across scientific domains, we processed 4,525 biomedical publications focusing on *Protein* (e.g. Myoglobin) entity type. Evaluation on the Craft corpus [11] shows that TSE-NER can achieve performance comparable to existing dictionary-based systems, and obtain precision up to 0.40 and recall up to 0.28 with just 25 seed terms. TSE-NER is implemented in the *SmartPub* platform [145]; its source code is available on Github[4], and its application shown in the video screencast at the following address: `https://youtu.be/zLLMwOT5sZc`.

Outline. The remainder of the chapter is organized as follows. In Section 5.2 we first briefly cover related work. Section 5.3 presents our approach, and describes alternative data expansion and entity filtering strategies. The experimental setup and results are presented in Section 3.4. Section D.6 concludes.

3.2 Related Work

A considerable amount of literature published in recent years addressed the *deep* analysis of text. Common approaches for *deep* analysis of publications rely on techniques such as bootstrapping [200], word-frequency analysis [191], probabilistic methods like Latent Dirichlet Allocation [75], etc. In contrast to current research [191] which limits the analysis of a publication's content to its title, abstract, references, and authors, we extract entity instances from the much richer full text. In addition, our method does not rely on existing knowledge bases [185, 166] and it is not based on selecting the most frequent keywords [191]. More recent research [192] used both corpus-level statistics and local syntactic patterns of scientific publications to identify entities of interest. Our method uses only a small set of seed names (i.e 5-100), and automatically trained distributed word representations to train a NER in iterative steps (i.e. 2-3).

[4] `https://github.com/mesbahs/TSE-NER/blob/master/README.md`

Entity Instances Extraction Named Entity Recognition (NER) has been applied to identify both entity types of general interest (e.g. Person, Location, Cell, Brand, etc.) as well as for specific domains (e.g., medicine or other domain where resources for training a NER are easily available). NERs rely on different approaches such as dictionary-based, rule-based, machine-learning [192] or hybrid (combination of rule based and machine learning) [205] techniques. Despite its high accuracy, a major drawback of dictionary-based approaches is that they require an exhaustive dictionary of domain terms, which are expensive to create and many smaller domains lack the resources to do so. The same holds for rule-based techniques, which rely on formal languages to express rules and require comprehensive domain knowledge and time to create.

Bootstrapping and Entity Set Expansion. Most current NERs are based on Machine Learning techniques, which require a large corpus of labeled training text [81]. Again, the high costs of data annotation is one of the main challenges in adopting specialized NER for rare entity types in specialized domains [192]. In recent years, many attempts have been made to reduce annotation costs. Active learning techniques have been proposed, asking users to annotate a small part of a text for machine learning methods [71].

Transfer learning techniques [172] use the knowledge gained from one domain and apply it to a different but related named entity type. Automatically create training data using seed list [24]. In contrast to previous work, we do not require a large training corpus [172] for transfer learning or a large seed list [24]; also, our approach differs from works on high-recall entity extractors (e.g. with regular expression extractors) for detecting entity types such as location and age [101]. Our focus is to augment the training data when only a small seed set of instances of the targeted entity type is available.

Entity Set Expansion is a technique finding similar entities to a given small set of seed entities [23, 101]. Bootstrapping [200] is another approach similar to our method that uses seed terms and extracts features such as unigrams, bigrams, left unigram, closest verb, etc. These are used to annotate more concept mentions which leads to extracting new features. This step operates in an iterative fashion until no new features are detected. Our approach is inspired by Entity Set Expansion and bootstrapping, but relies on different expansion strategies and does not require concepts already being available in knowledge bases [23].

3.3 Approach

Our TSE-NER (Term and Sentence Expansion) approach for domain-specific long-tail entity recognition is organized in five steps, as shown in Figure 3.1.

❶ An initial set of seed terms is used to identify a set of sentences used as initial *training data* (Section 3.3). ❷ *Expansion* strategies can be used to expand the set of initial seed terms, and the *training data* sentences (Section 3.3). ❸ The *Training Data Annotation* step annotates the training data using the (possibly expanded) seed terms set (Section 3.3). ❹ A new Named Entity Recognizer (NER) is *trained* using the annotated training data, and the newly trained NER is applied on the corpus to detect a candidate

set of entities (Section 3.3). ❺ The *Filtering* step refines the set candidate entities set, to improve the quality of outputted *Verified Terms* set (Section 3.3).

TSE-NER operates under the hypothesis that there are recurring patterns in the mentions of domain-specific named entities, and that they appear in similar contexts. If this hypothesis holds, by training a classifier on the texts containing the entities, we are able to extract the instances of the entity type of interest. The process can be iterated, by repeating the first step using the newly detected terms as seeds to generate new training data. We rely on the following concepts (some are only relevant for the evaluation, and could be omitted in setups where evaluation is not necessary).

Known Entity Terms $T_{all} := T_{seed} \cup T_{test}$: This represents a manually created set of instances of the entity type for which a NER classifier is to be trained. In this work, we split this set into a set of seed terms T_{seed} used for training, and test terms T_{test} used for evaluation purposes. In a real-life scenario not requiring a formal evaluation, of course only the seed terms would be necessary. T_{seed} may be small. In this work we consider seed sets $5 \leq |T_{seed}| \leq 100$. Creating T_{seed} is the only manual input required for NER training in our approach.

Document Corpus $D_{all} := \{d_1, ..., d_{|D|}\}$: This is the complete document corpus available to our system. Parts of it can potentially be used for training, others for testing. Each document is considered to be a sequence of sentences.

All Sentences $S_{all} := \{s | s \in d \wedge d \in D_{all}\}$: This represents all sentences of the whole document corpus. Each sentence is considered to be a sequence of terms.

Test Sentences $S_{test} := \bigcup_{t \in T_{test}} \{s | s \in S_{all} \wedge t \in s\}$: These are all sentences containing any term from the test set, and they need to to be excluded from any training in order to ensure the validity of our later evaluations, resulting in the set of **Development Sentences** $S := S_{all} \setminus S_{test}$.

In the following, we introduce the iterative version of our approach, representing the current iteration number as i whereas initially $i = 0$. Each iteration i uses its own term list T_i, which initially is $T_0 \subseteq T_{seed}$ (the size of the subset of T_{seed} depends on the desired use case, as discussed in section 3.4).

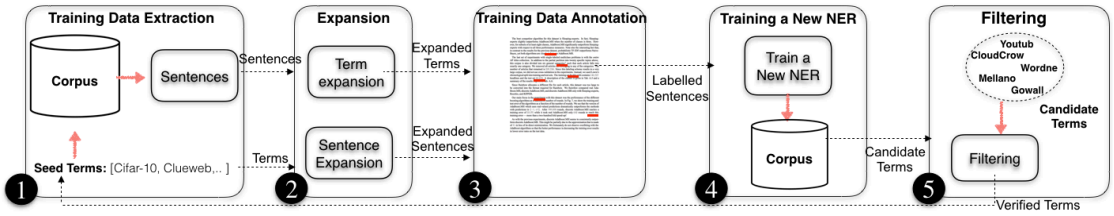


Figure 3.1: Overview of the domain-specific long-tail named entities recognition approach.

Training Data Extraction

As a first step, a set of training data sentences S_i for the current iteration is created by extracting suitable sentences from S. At this stage, this is realized by selecting all sentences containing any of the seed terms. Therefore, S_i provides examples of the positive classification class as they are guaranteed to contain a desired entity instance. To better capture the usage context of the seed entity, we also extract surrounding sentences in the text: $S_i := \cup_{t \in T_i}\{s | s \in S \wedge (t \in s \vee t \in successor(s) \vee t \in predecessor(s))\}$.

Expansion

The small size of the seed term set T_{seed} has two obvious shortcoming that can greatly hinder the accuracy and recall of the trained NERs: 1) the amount of training data sentences S_i is limited; and 2) there are only few examples of mentions of the entity instances of the given type. In addition, the *generalization* capability of the NER for identifying new named entities can also be affected: an insufficient amount of positive examples can lead to entities of the targeted type being labeled negatively; while the extraction of sentences in the training data that are related to seed terms will cause a shortage of negative examples. To account for these issues, we designed two *expansion* strategies.

Term Expansion (TE). Term Expansion is designed to increase the number of known instances of the desired entity type before training the NER. An expanded set of entities will provide more positive examples in the training data, thus ideally improving the precision of the NER. In scientific documents, it is common for domain-specific named entities to be in close proximity, e.g. to enumerate alternative solutions, or list technical artifacts. The *Term Expansion* (TE) strategy is therefore designed to test and exploit this hypothesis.

We introduce the interface $expandTerms(terms_s)$, with $terms_s \subseteq terms_i$. While many different implementations for this interface are possible, in this work we use *semantic similarity*: terms which are semantically similar to terms in the seed list should be included in the expansion. For example, given the dataset seed terms `Clueweb` and `cim-10`, the expansion should add similar terms like `trec-2005`.

We exploit the distributional hypothesis [84] stating that terms frequently occurring in similar context are semantically related, using the popular *word2vec* implementation of skip-n-gram word embeddings [153]. In essence, *word2vec* embeds each term of a large document corpus into low-dimensional vector space (100 dimensions in our case), and the cosine distance between two vectors has been shown to be a high-quality approximation of semantic relatedness [126]. In our implementation, we trained the *word2vec* model on the whole development sentence collection S, as described in [153], learning all uni- and bigram word vectors of all terms in the corpus. Then, in its most basic version, we select all terms from all sentences, and cluster them with respect to their embedding vectors using K-means clustering. Silhouette analysis is used to find the optimal number k of clusters. Finally, clusters that contain at least one of the seed terms are considered to (only) contain entities the same type (e.g *Dataset*).

Algorithm 1 TE using Semantic Relatedness

function EXPANDTERMS($terms_s$)
$T_{entity} := \{t | t \in s \wedge s \in S \wedge isEntity(t)\}$
▷ All entities in S
$clusters := cluster(word2vec(T_{entity}))$
▷ Cluster the embeddings
$clusters_{correct} := \{c | c \in clusters \wedge t \in terms_s \wedge t \in c\}$
▷ Select clusters containing any initial term
return $\bigcup_{c \in clusters_{correct}}$

Initial experiments have shown that this naive approach is slow, and that it can potentially introduce many false positives due to 1) the large number of considered terms, and 2) the sometimes faulty assumption that all terms in cluster are indeed similar as *word2vec* relatedness is not always reliable for similarity measurements [126]. To improve, in the following we only consider terms terms which are likely to be named entities by using NLTK entity detection to obtain a list of all entities E_{all} contained in S.[5] This results in the Algorithm 1.

Sentence Expansion (SE). A second (optional) measure to increase the size and variety of the training set in a guided fashion is the *Sentence Expansion* (SE) strategy (shown in Algorithm 2). It addresses the problem of the over-representation of positive examples resulting from selecting only sentences with instances of the desired type (see section 3.3). The goal is to include sentences which are unlikely to contain instances of the desired type, but are still very similar in semantics and vocabulary to serve as informative negative examples in order to boost the NER training accuracy.

For this, we rely on *doc2vec* document embeddings [111], a variant of *word2vec*, to learn vector representations of the sentences in the corpus. For each sentence in the development set, we use *doc2vec* to discover the most similar sentence which does not contain any known instance of the targeted type (i.e., expanded terms). While indeed such sentences sometimes do contain an unknown instance of the targeted entity type, which would now be misclassified in the training set). To minimise such possibility, in our experiments sentence expansion always include the term expansion strategy.

Algorithm 2 Optional Sentence Expansion

function EXPANDSENTENCES(S_{org})
return $sentences \cup \{s | s' \in S_{org} \wedge s \in S \wedge mostSimilar_{doc2vec}(s, s')\}$

[5]NLTK entity detection is based on grammatical context. It does not perform any typing, and due to it's simplicity, has high recall values.

Table 3.1: Stanford NER training parameters.

useWord=true	useLastRealWord=true
useNGrams=true	useNextRealWord=true
usePrev=true	lowercaseNGrams=true
useNext=true	featuresuseTypeSeqs=true
useLemmas=true	useTypeSeqs2=true
normalize=true	useTypeySequences=true
useOccurrencePatterns=true	wordShape=chris2useLC

Training Data Annotation

After obtaining an (expanded) set of instances T_i (the current term list) and training sentences S_i, we annotate each term $A_{T_i} := annotate_{T_i}(S_i)$ in all training sentences if they are a positive instance of the targeted entity type, i.e. if the term $\in T_i$. Using A_{T_i}, any state-of-the-art supervised NER can be trained.

NER Training

For training a new NER_i, we used the Stanford NER tagger[6] to train a Conditional Random Field (CRF) model. As the focus of this chapter is the process of training data generation, we do not consider additional algorithms. CRF has shown to be an effective technique on different NER tasks [110]; the goal of CRF is to learn the hidden structure of an input sequence. This is done by defining a set of feature functions (e.g. word features, current position of the word labels of the nearby word), assigning them weights and transforming them to a probability to detect the output label of a given entity. The features used in the training of the model are listed in Table 3.1. After a NER for the current iteration N_i is trained, it is used to annotate the whole development corpus S, i.e. $A_{NER_i} := annotate_{NER_i}(S)$. All positively annotated terms are considered newly discovered instances of our desired type.

Filtering

After applying the NER to the development corpus, we obtain a list of new candidate terms. As our process relied on several steps which might have introduced noise and false positives (like the expansion steps, but also the NER itself), the goal of this last (optional) step is to filter out candidate terms that are unlikely of the targeted type using a set of external heuristics with different assumptions:

Wordnet + Stopwords (WS) Filtering. In the domain-specific language of scientific documents, it is common for named entities to be "proper" of that domain (like `Simlex-999`), or to be expressed as acronyms (like `Clueweb`, `SVM`, `RCV`). In this strategy, named entities are assumed to be not relevant if they are part of the "common" English language,

[6] `https://github.com/dat/stanford-ner`

either as proper nouns (e.g. *software*, *database*, *figure*), or a Stopwords (e.g. *on*, *at*). This is achieved by performing lookup operations in WordNet[7] and in common lists of stopwords.[8] As both sources focus on general English language, only domain-specific terms should be preserved.

Similar Terms (ST) Filtering. In order to distinguish between different entity types that pertain to a given domain (e.g. `SVM` is of type `Method`, while `Clueweb` is of type `Dataset`), this filtering strategy employs an approach similar to the one used in the *Term Expansion* (TE) strategy. The idea is to cluster entities based on their embedding feature using K-means clustering, and keep all the entities that appear in the cluster that contains a seed term.

Pointwise Mutual Information (PMI) Filtering. This filtering strategy adopts a semantic similarity measure derived from the number of times two given keywords appear together in a *sentence* in our corpus. The heuristic behind this filter is vaguely inspired by Hearst Patterns [187], as we manually compile a list of context terms / patterns CX which likely indicate the presence of an instance of our desired class (e.g., "we evaluate on x" typically indicates a dataset). Unlike the other filters, it does increase the manual resource costs for training.

Given a set of candidate entities CT_i and the context term set CX, we measure the PMI between them using $\log \frac{\mathrm{N(ct,cx)}}{\mathrm{N(ct)N(cx)}}$ with $ct \in CT_i \wedge cx \in CX$, and $N(ct, cx)$ being the number of sentences in which both a candidate entity (ct) and a given keyword (t) occur (analogously, $N(ct)$ counts the number of occurrences of ct). Finally, candidate terms are filtered and excluded if their PMI value is below a given threshold value.

Knowledge Base Lookup (KBL) Filtering. Our target are long-tail domain-specific entities, i.e. entities that are not part of existing knowledge bases. Named entities that could be linked to a knowledge base could be assumed incorrect, and therefore amenable to exclusion from the final named entity set. In the KBL approach we exclude the entities that have a reference in the DBpedia.

Ensemble (EN) Filtering. Different filtering strategies are likely to remove different named entities. To reduce the likelihood of misclassification, the *Ensemble* (EN) filtering strategy combines the judgment of multiple filtering strategies, to preserve candidate entities that are considered correct by one or more strategy. Intuitively, if each strategy makes different errors, then a combination of the filters' judgment can reduce the total error. We preserve the entities that are passed through two out of three selected filtering strategies.

Summary

In Algorithm 3, we summarize the previous subsections into a unified algorithm covering the whole iterative NER training workflow.

[7] `http://wordnet.princeton.edu/`

[8] `http://www.nltk.org/book/ch02.html`

Algorithm 3 Iterative NER Training

function LONGTAILTRAIN(T_{seed}, S_{all})
  $T_0 := T_{seed}$
  for $i \in \mathbb{N}_0$ **do**
    $S_i := \cup_{t \in T_i}\{s | s \in S \wedge (t \in s \vee t \in successor(s) \vee t \in predecessor(s))\}$
    $T_i := expandTerms(T_i)$
    (optional) $S_i := expandSentences(S_i)$
    $A_{T_i} := annotate_{T_i}(S_i)$
    $NER_i := trainNER(A_{T_i})$
    $A_{NER_i} := annotate_{NER_i}(S)$
    $CT_i := isPositiveTermIn(A_{NER_i})$
    $FT_i := filter(CT_i)$
    $T_{i+1} := FT_i \cup T_{seed}$
    if convergence **then**
      return A_{NER_i}

3.4 Evaluation

This section reports on an empirical evaluation to assess the performance of the approach (and its variants) described in Section 5.3, and the ability to utilize it for long-tail named entity recognition. Section 3.4 describes the experimental set-up, followed by the results (Section 3.4), and their discussion (Section 3.4).

Experimental Setup

Corpora. Our main evaluation, shown in the following sections, is performed on the data science (15,994 papers from 10 conference series) domain. To assess the performance of TSE-NER in other scientific domains, at the end of the section we describe an experiment over 4,525 publications from 10 biomedical journals. The full description of the corpora is described in the Github page [9].

Publications are processed using GROBID [129], to extract a structured full-text representation of their content.

Long tail entity types selection. Scientific publications contain a large quantity of long-tail named entities. Focusing on the data science domain, we address the entity types *Dataset* (i.e. dataset presented or used in a publication), and *Methods* (i.e. algorithms – novel or pre-existing – used to create/enrich/analyze a dataset). Both entities types are scarcely represented in existing knowledge bases.[10] To evaluate the performance of our approach, we create a set of 150 seed instances T_{all} for each targeted type, collected public from public websites.[11]

[9] https://github.com/mesbahs/TSE-NER/blob/master/README.md

[10] In DBPedia, the type `dbo:database` features 989 instances, but mostly related to biology, economy, and history. The type `dbo:software` contain names of several algorithms, but the list is clearly incomplete.

[11] For instance: https://github.com/caesar0301/awesome-public-datasets. The full list of seed entity instances,

Table 3.2: Size Statistics of various seed set sizes #S

Entity type	***#S***	***#Sentences***	***#Words***
Dataset Training	5	198	2081
	10	358	4737
	25	799	13080
	50	1456	29015
	100	2863	63517
Method Training	5	617	15682
	10	1192	30354
	25	3620	86563
	50	7910	190026
	100	18543	449515
Dataset Testing	50	3149	69272
Method Testing	50	1097	26426

For each type, 50 of those are selected as test terms for that type T_{test}, while 100 are used as seed terms T_{seed}.

Evaluation Dataset. As discussed in Section 3.3, in the training process all test sentences S_{test} (i.e. sentences mentioning terms in T_{test}) in the corpus D_{all} are removed. For evaluation, we manually created a type-annotated test set: for each test term, we select all sentences in which they are contained including any adjacent sentence, forming the set of annotated sentences $S_{annotated} := \cup_{t \in T_{test}} \{s | s \in S_{test} \wedge (t \in s \vee t \in successor(s) \vee t \in predecessor(s))\}$. An expert annotator labeled each term as an instance of the target type to create the test annotation set used for evaluation $A_{test} := annotate_{expert}(S_{annotated})$.

Details of statistics on sentences used for training and testing can be found in Table 3.2. For training, depending on the seed set size between 5 and 100, we used between 198 and 2863 sentences for the *dataset* entity type and 617 to 18545 sentences for the *Method* entity type.

Algorithm 4 Evaluation Protocol

function EVALUATE($seed\_size$)
  $T \subseteq_{seed\_size} T_{seed}$
  $NER_{final} := longtailTrain(T, S_{all})$
  $A_{final} := annotate_{NER_{final}(S_{annotated})}$
  $result := analyze(A_{final}, A_{test})$

For testing 50 seed terms were used for both dataset (i.e. 3149 sentences) and method (i.e. 1097 sentences) entity type. The evaluation protocol is described in Algorithm 4, where the $seed\_size$ values can be initialized with different values. Our analysis was not limited to the 50 test seed terms, we further evaluated 200 entities recognized by TSE-NER via a pooling technique.

as well as the list of sources are available on the Github page`https://github.com/mesbahs/TSE-NER/blob/master/README.md`.

Results

For a given entity type (*Dataset* and *Method*), we test the performance with differently sized seed sets and expansion strategies to create the training data for generating the NER model, and different filtering strategies to filter the resulting set of recognized entities. We report the performance of the basic WS, PMI, and EN strategies, plus a combination of the WS, ST, and KBL strategies, as listed in the caption of Table 3.3. We also perform an experiment to test the performance of our approach when applied iteratively. We analyze the performance of the model on the manually annotated test set presented in the previous section.

Tables 3.3 and 3.4 summarize the performance achieved for *Dataset* and *Method* entity types. In Table 3.4, the *No Expansion* and *Term Expansion* figures for the *Method* type are omitted for brevity's sake. Our approach is able to achieve excellent precision [89% – 91%] with both entity types, and good recall (up to 41%) with the *Dataset* type. The lower recall obtained with the *Method* type can be explained with the greater diversity (in terms of n-grams and use of acronyms) of method names.

The expansion strategies lead to an average +200% (SE – *Dataset*) and +300% (TE – *Dataset*) increase in recall, thus demonstrating their effectiveness for generalization. On average, filtering decrease recall, but with precision improvements up to +20% (PM – *Method*). These are promising figures, considering the minimal human supervision involved in the training of the NERs. We can also show the different trade-offs our approach can strike: different configurations of filtering and expansion lead to different results with respect to precision and recall values, allowing for example a high-precision slightly-lower recall setup for a digital library, and a higher recall lower precision setup for a Web retrieval system.

Expansion Strategies. Expansion strategies increase the size and variety of training datasets, thus improving the precision and recall. Both strategies achieve the expected results, although with different performance increase: compared to *NE* strategy, both TE and SE achieve a considerable performance boost ($\mu = +190\%$) for recall, but at cost of lower precision ($\mu = -8.7\%$). We account the better recall performance of TE to the contextual similarity (and proximity) of named entities of the same type in technical documents (e.g. `Gov2`, `Robust04`, `ClueWeb` and `Wt10g`). The precision decrease in TE can be accounted to treating some terms incorrectly as positive instances due to their presence in the same embedding clusters as the seed terms (see also Section 3.3). The SE strategy shows lower recall ($\mu = +210\%$ over NE), but with less precision loss ($\mu = -5.2\%$ than NE). We account this positive behaviour to the presence of more quality negative examples, helping to maintain the generalization capabilities of the NER, while refining the quality of its recognition.

Filtering Strategies. We observe no significant improvement in precision with the WS filtering approach. Manual inspection of results reveal that most of the false positives are already domain-specific terms (e.g. `Pagerank`, `Overcite` for *Dataset*, and `NDCG` for *Method*) which are not included in Wordnet, but that are of the wrong type. SS slightly increases the precision by keeping only the entities that appear in the same cluster as the seed

names; however, this comes at a cost, as the recall is also penalized by the exclusion of entities of interest that are in other clusters. KB excludes popular entities that are contained in the knowledge base (e.g. `Wordnet`, `Dailymed`), but also some rare entities that are mistyped.

Table 3.3: *Dataset* entity type: Precision/Recall/F-score on evaluation dataset. Legend: *NE* – No Expansion; *TE* – Term Expansion; *SE* – Sentence Expansion; *NF* – No Filtering; *WS* – Wordnet + StopWords; *SS* – Similar Terms + WS; *KS* – Knowledge Base Lookup + SS; *PM* – Point-wise Mutual Information; *EN* – Ensemble.

Strategy	***#S***	***NF***	***WS***	***SS***	***KS***	***PM***	***EN***
NE	5	.83/**.05**/**.10**	.84/.04/.08	.86/.03/.07	.75/.01/.01	**.90**/.04/.09	.86/.04/.08
	10	.84/**.07**/**.14**	.83/.06/.12	.85/.06/.11	.78/.01/.02	**.90**/**.07**/.13	.85/.06/.11
	25	.84/**.08**/**.16**	.83/.07/.13	.86/.07/.13	.78/.01/.03	**.91**/**.08**/.15	.85/.07/.13
	50	.85/**.12**/**.21**	.84/.10/.18	.87/.10/.18	.80/.02/.05	**.92**/.11/.20	.86/.10/.18
	100	.85/**.15**/**.26**	.85/.13/.22	.87/.12/.22	.82/.03/.07	**.91**/.13/.24	.86/.12/.22
TE	5	.76/**.14**/**.25**	.78/.13/.22	.79/.11/.20	.74/.04/.09	**.83**/.13/.23	.80/.13/.22
	10	.72/**.24**/**.36**	.74/.21/.33	.76/.21/.33	.70/.10/.18	**.78**/.22/.35	.76/.21/.33
	25	.72/**.29**/**.42**	.73/.28/.40	.75/.27/.40	.73/.17/.28	**.77**/.27/.40	.75/.27/.40
	50	.70/**.36**/**.47**	.71/.33/.46	.73/.33/.45	.71/.21/.33	**.75**/.33/.46	.73/.33/.45
	100	.69/**.41**/**.51**	.70/.39/.50	.71/.38/.50	.71/.28/.40	**.74**/.38/.50	.72/.38/.50
SE	5	.83/**.07**/**.14**	.84/.06/0.12	.86/.05/.10	.82/.01/.02	**.91**/**.07**/.13	.86/.06/.11
	10	.81/**.15**/**.26**	.81/.13/.22	.84/.12/.21	.73/.02/.05	**.89**/.14/.25	.84/.12/.21
	25	.81/**.22**/**.35**	.80/.18/.29	.83/.17/.29	.77/.04/.08	**.89**/.20/.33	.82/.18/.29
	50	.78/**.27**/**.40**	.78/.22/.35	.81/.21/.34	.76/.06/.11	**.87**/.24/.38	.80/.22/.34
	100	.77/**.30**/**.43**	.77/.24/.37	.80/.23/.36	.78/.07/.13	**.86**/.26/.40	.79/.24/.37

Table 3.4: *Method* entity type: Precision/Recall/F-score. Legend as in Table 3.3.

Strategy	***#S***	***NF***	***WS***	***SS***	***KS***	***PM***	***EN***
SE	5	.76/**.04**/**.08**	.77/.03/.07	.77/.01/.01	.84/.01/.01	**.86**/.01/.03	.84/.03/.05
	25	.77/**.14**/**.24**	.77/.12/.21	.79/.09/.16	**.87**/.05/.09	.86/.05/.09	.85/.09/.17
	100	.68/**.15**/**.25**	.67/.14/.23	.65/.12/.20	.84/.07/.13	**.85**/.05/.10	.83/.10/.19

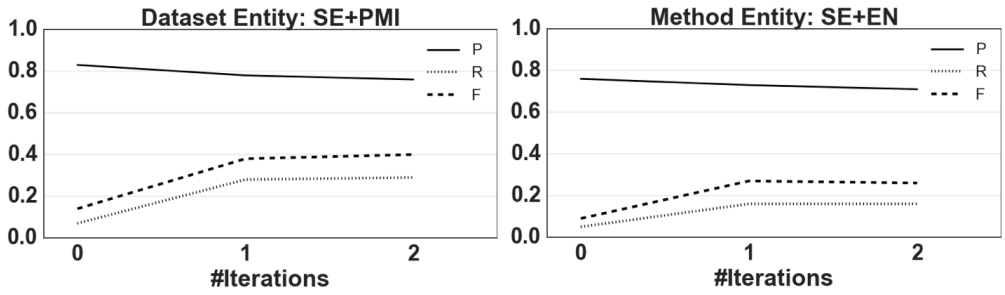


Figure 3.2: Dataset(L) and Method(R) entity: Iterative NER training using 5 initial seeds.

For instance, the *Dataset* entities `Ratebeer`[12] or `Jester` can be retrieved from DBpedia using the lookup search, although the result points to another entity. This is a clear limitation with the adopted lookup technique, which could be avoided with a more precise implementation of the lookup function. PMI usually gets the highest precision; the strategy proved effective in removing false positives, but penalizes recall by excluding entities that do not appear with the words in the context list CX. For instance, `Unigene` (*Dataset*) often appears in with the term *data source*, which is not in our context list and thus filtered out. The EN strategy keeps only the entities that are preserved by two out of three (WS, KB and PMI) filtering strategies. While reducing the number of false positives, this proves to be too restrictive; for instance *Dataset* names such as `Yelp`, `Twitter`, `Foursquare` and `Nasdaq` are removed by both the WS and KB strategies.

Seed Set Size. We randomly initialize $T \subseteq T_{seed}$ with $|T| = 5, 10, 25, 50, 100$ (see Algorithm 4). We execute the evaluation cycle 10 times for each size of T, and again vary expansion and filtering strategies. The recall performance sharply increase with the number of seeds term ($\mu = +340\%$ from 5 to 100 seeds): this is due to the increase in the number of sentences available for NER training, and is an expected behaviour. The decrease in precision is an average of -6% from 5 to 100 seeds, with an average value of -5.1% for *Dataset* and -6.9% for *Methods*. Noteworthy are the good performance with as little as 5 seed entities (*Datasets*: 0.25 F-score with TE strategy and no filtering).

Iterative NER Training. Figure 3.2 shows the result of the iterative NER training using Sentence Expansion with 5 seeds. We report the results with the PMI (*Dataset*) and EN (*Methods*) filtering, as they are the ones offering the most balanced performance in both precision and recall. Despite the small initial seed seed, it is possible to achieve precision and recall comparable to the ones obtained with an initial set of 100 seeds in only 2 iterations.

Analysis of recognized entities. To widen the scope of our evaluation, we extended our result analysis beyond the 150 named entities in T_{all}. We manually investigated up-to-now unknown named entities which have been recognized by the NER after training. We applied a method inspired by the pooling technique typically used in information retrieval research: given a list of seed terms T_{seed} of a given type, and a list of recognized

[12] `http://lookup.dbpedia.org/api/search/KeywordSearch?QueryClass=&QueryString=ratebeer`

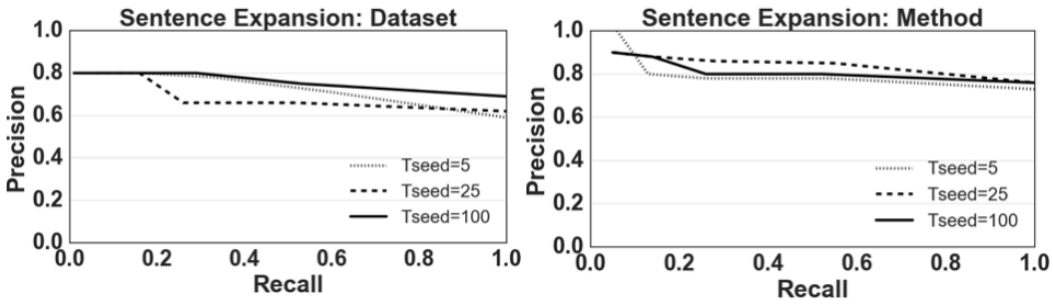


Figure 3.3: *Dataset* (L) and *Method* (R): Precision and Recall for ranked top 10, 25, 50, 100 and 200 entities, varying seeds sizes.

potential filtered terms FT of an yet unknown type, the idea is to rank the items in the list of candidate terms FT according to their embedding similarity to the items in the seed set T_{seed} and collect the top K. As a result, the obtained precision and recall measurements are only approximate values. The similarity is measured based on the cosine similarity between the *word2vec* embedding vectors. Each entity in the lists has been manually checked by an expert. Figure 3.3 shows the precision and recall of the top $K = 10, 25, 50, 100$, and 200 retrieved entities using the SE approach. As in the previous experiment, we used the PMI and EN filtering strategies respectively for *Dataset* and *Method* types. Precision performance are consistently high at all level of recall.

The *Dataset* entities `mslr-web10` (a benchmark collection for learning to rank method) and `ace2004` (ACE 2004 Multilingual Training Corpus); and *Method* entities such as *TimedTextTank* and *StatSnowball* are a sample of extracted entities. Some examples of incorrect detected entities are due to ambiguous nature of the sentence. Consider the following sentence: *"The implementation of scikitlearn toolkit was adopted for these methods"*, since it is similar to a sentence that contains a method entity, the entity *scikitlearn* was detected as a method although its a software library. In another sentence: *"The Research Support Libraries Programme (RSLP) Collection Description Project developed a model."*, *RSPL* (a project) was detected as a dataset due to its surrounding words (e.g. *collection*, *libraries*).

Comparison with State-of-the-art. We compared our method with: 1) the Boot-Strapping (BS) based concept extraction approach [200], a commonly used state-of-the-art technique in scientific literature; the experiments where executed with the code and the parameters (k, n, t) to (2000, 200, 2) provided in [200], and with 100 seeds. And, 2) improved and expanded Hearst Pattern (HP) [187] for automatically building or extending knowledge bases extracting type-instance relations e.g., X such as Y as in "we used datasets such as twitter". Intuitively, the performance of BS decreases with less number of seed terms. For the HP we kept type-instance pairs related to dataset or method (i.e. the context words in CX). Experiments on our evaluation dataset shown that TSE-NER achieved higher performance in terms of precision/recall/fscore for the *dataset* entity type (0.77/0.30/0.43) compared to *BS* (0.08/0.13/0.10) and *HP* (0.92/0.15/0.27) as well as for the *method* entity (*TSE-NER*: 0.68/0.15/0.25, *BS*: 0.11/0.32/0.16, *HP*: 0.64/0.04/0.07).

The high precision and low recall in *HP* is explained by the limited set of *HP* patterns. We infer that different expansion strategies augment the performance of our technique compared to the *BS* which just relies on features such as unigrams, bigrams, closest verb, etc.

Biomedical Domain. To test the performance of TSE-NER on another scientific domain, we processed 4,525 biomedical publications from 10 journals focusing on the *Protein* entity type. The seed terms were selected from the protein ontology.[13] We excluded the test terms appearing in the Craft corpus [11] (a manually annotated corpus containing 67 full-text biomedical journals) and kept only the ones that have a reference in the publications. The list of seed terms used for the *Protein* entity are listed on the Github page [14]. We randomly initialized $T \subseteq T_{seed}$ with $|T| = 5, 25, 100$ and employed the SE strategy and a simple WS filtering. The evaluation cycle has been executed 10 times for each size of T, and results are averaged. TSE-NER can achieve precision/recall/f-score of 0.57/0.08/0.14 using 5 seeds, 0.40/0.28/0.32 using 25 seeds, and 0.38/0.46/0.41 with 100 seeds. The latter results are comparable to extensive dictionary-based systems [203] (0.44/0.43/0.43) [65] (0.57/0.57/0.57) where existing ontologies in the biomedical domain are used for matching *Protein* entities of the text.

Discussion

The design goal of our TSE-NER approach was minimizing the training costs in scenarios where the targeted entity types are rare, and little to no resources (for manual annotations) are available. In these cases, relying on exhaustive dictionaries or knowledge-bases is not possible, and common techniques like supervised learning cannot be applied. We believe to have successfully reached that goal, as we could show that even with small seed lists T_{seed} with little as 5 or 25 terms, high-precision NERs could be trained.

Nonetheless, this ease-of-training comes at a price: recall values are low, and are unlikely to be able to compete with known much more elaborately trained NERs for popular types. However, by selecting different configurations for filtering and expansion, recall can be moderately improved at the cost of precision. Also, the effectiveness of such changes of configurations seems to slightly differ between the *Dataset* and *Method* entity types. As a result, we cannot identify one clear best configuration as TSE-NER seems to benefit from some entity type-specific tuning. However, this also provides some flexibility to tune with respect to different quality and application requirements.

Furthermore, some of our underlying assumptions, heuristics and implementation choices, are designed as a simplistic proof-of-concepts, and deserve further discussion and refinement. As an example, consider WS WordNet filtering: we assumed domain-specific named entities would not be part of common English language. While this is true for many relevant domain-specific entities, several datasets (for instance) do indeed carry common names like the `census dataset`. For a production system, more complex implementations and tailored crafting is necessary for reaching higher performance values.

[13] `http://obofoundry.org/ontology/pr.html`

[14] `https://github.com/mesbahs/TSE-NER/blob/master/README.md`

Another restriction is related to the core heuristics found in the term and sentence expansion, where we assume that similar types of entities occur in similar contexts – which is not necessarily always the case.

Threats To Validity. Our evaluation has been performed on an extensive document corpus, covering two distinctively different domains. However, we focused only on a limited set of entity types. The hypothesis described in Section 5.3 hold for *Datasets*, *Methods*, and *Proteins*, but further experiments are needed for other entity types in the same domains (e.g. *Software*) or in other domains. Despite the good performance achieved, it could already be noted that even between those three types, no single TSE-NER configuration is clearly the best. In order to obtain a complete understanding of the full capabilities, limitations, and trade-offs of our approach, more studies addressing additional domains and entity types are necessary.

3.5 Conclusion

We presented a novel approach for augmenting training data for Long-tail Entity Recognition (L-tER). A limiting factor in this scenario is the lack of resources or available explicit knowledge to allow for established NER training techniques. We explored techniques able to limit the reliance on human supervision, resulting in an iterative approach that requires only a small set of seed terms of the targeted type. Our core contributions, in addition to the overall approach, are a set of expansion strategies exploiting semantic similarity and relatedness between terms to increase the size and labeling quality of the training dataset generated from the seed terms, as well as several filtering techniques to control the noise introduced by the expansion.

In our evaluation, we could show that we can reach a precision of up to 0.91, or a recall of up to 0.41 – a good result considering the very cheap training costs. Furthermore, we could show that recall can be traded for more precision to a moderate extent by changing the configuration of our NER training process.

For future work, additional evaluation addressing more domains and entity types is of importance to better understand the range of applicability of our approach. Also, many of our currently still simplistic heuristics and implementation choices can benefit from improvement and optimization. This leads us to our next chapter, where we try to incrementally incorporate human feedback on the relevance of extracted entities into the training cycle of such iterative TSE-NER algorithms to improve the overall performance concerning precision, recall, and F-measures.

Chapter 4

A Collaborative Approach for improving the Extraction and Typing of Long-tail Entities

With the work presented in Chapter 3, we observed that the presented approach achieved promising results relying on training NER techniques in an iterative fashion, thus limiting human interaction to only providing a small set of seed terms. The approach heavily relied on heuristics in order to cope with the limited training data size. As these heuristics are prone to failure, the overall achievable performance is limited. In this chapter we address RQ3 by introducing a collaborative approach that incrementally incorporates human feedback on the relevance of extracted entities into the training cycle of such iterative NER techniques. This approach, called Coner, allows to still train new domain-specific rare long-tail NER extractors with low costs, but with ever increasing performance while the algorithm is actively used in an application. The contribution of this chapter is published in [210].

4.1 Introduction

With the ever increasing amount of scientific publications, there is a growing need for methods that facilitate the exploration and analysis of a given research field in a digital library collection [133], but also for techniques which can provide effective retrieval and search experiences. To this end, *"deep meta-data"* extracted from scientific publications allows for novel exploration capabilities [141].

Domain-specific typed named entities [143] are a representative example of deep meta-data. Consider the domain of *data processing and data science*, which is currently popular due to its real-life implications on machine learning algorithms and data-centric business models. In this domain, the main entity types of interests to the user base of a scientific collection would for example be: *datasets* used in a given publication; the *methods* applied to the data or used in implementation; or *software packages* realizing these methods [138]. However, extracting and typing named entities for this scenario is hard, as most entities relevant to a specific scientific domain are very rare, i.e. they are part of the *entity long-tail*. Most current state-of the art Named Entity Recognition (NER) algorithms focus on high-recall named entities (e.g., locations and age) [101], as they rely on extensive manually curated training and test data. Due to the rare nature of long-tail entity types, training data is scarce or non-available. Some approaches addressed this problem by relying on bootstrapping [200] or entity expansion [23, 101] techniques, achieving promising performance. However, how to train high-performance *long-tail* entity extraction and typing with minimal human supervision remains an open research question.

Recently, TSE-NER [147] was presented, an iterative approach for entity extraction in scientific publications. The approach starts with a small seed set of known entity instances; for each type it is sufficient to have one or two domain experts denote between 5 to 50 known entities. These sets are then heuristically expanded and annotated to generate training data to train a new traditional NER classifier, and heuristically filtered to remove likely false positives to create the entity set for the next iteration. As results of experiments in [147] have shown, this approach is hampered by the simplicity and unreliability of the heuristics used for expanding, but especially by those used for filtering the current iteration's entity set. Nonetheless, the approach promises a lot of potential if these heuristics can be improved.

The core goal of this chapter is to extend TSE-NER with incremental, collaborative feedback from human contributors to support the heuristic filters. We introduce `Coner`, an approach that allows the users of our system to continuously provide easy-to-elicit low-effort feedback on the semantic fit and relevance of extracted . Also, new entities may be added that they deem relevant for a specific facet / type. This feedback is then exploited to support the heuristic expansion and filter phases of the TSE-NER algorithm. The human-in-the-loop approach allows us to still maintain the advantages of the initial design of TSE-NER (i.e., training a NER algorithm cheaply, only relying on a small seed set, and providing an immediate result to users with acceptable extraction quality as discussed in [147]), while exploiting the human feedback into the next NER training

iteration. Coner allows the TSE-NER system to improve its performance over time by benefitting from additional human intelligence in the training process.
The contribution of this chapter are as follows:

- We describe `Coner`, an extension for TSE-NER which incorporates collaborative user feedback for continuously supporting the term expansion and entity filtering steps. The Coner pipeline consists of two novel modules: a document annotation viewer that visualises named entities and allows users to interact with them, and a feedback analyser that calculates relevance scores for evaluated entities and integrates them into TSE-NER heuristics. Coner is available as an open-source project.[1].
- We performed two experiments to evaluate our approach on a collection of 11,589 data science publications from ten conference series: 1) an exploratory experiment performed on 10 papers and with 10 users showing that by utilizing human feedback, up to **94.3%** of false positives can be detected for the *dataset* entity type and **57.9%** for the *method* entity type; 2) similar to experiment (1) but receiving only human feedback on entities with high expected information gain in order to maximize the impact of user feedback. This resulted in an average per-entity annotation time of just above 15 seconds and an increase of precision of up to 4% by boosting the expansion and filtering steps of TSE-NER.

4.2 TSE-NER: Distantly Supervised Long-tail NER

In this section we will summarize TSE-NER, an iterative five-step low-cost approach for training NER/NET classifiers for long-tail entity types. For more detailed information on this approach, refer to [147]. The approach is summarized in the following five steps:

1. For *Training Data Extraction*, a set of *seed terms* is determined, which are known named entities of the desired type. The *seed terms* are then used to identify a set of sentences containing the term.
2. *Expansion strategies* are used to automatically expand the set of seed terms of a given type, and the training data sentences.
3. The *Training Data Annotation* step is used to annotate the expanded *training data* using the expanded seed terms.
4. A new *Named Entity Recognizer* (NER) will be trained using the annotated training data for a the desired type of entity.
5. The *Filtering step* refines the list of extracted named entities by heuristically removing those entities which are most likely false positives. The set of remaining entities is treated as a seed set for the next iteration. This step is the focal point of this chapter.

[1] `http://removedForAnononymity.edu`

Training Data Extraction

In the first step, a set of training data sentences is created by extracting all the sentences containing any of the seed terms. In the first iteration, the seed term set can contain from 5 to 50 terms. They are provided manually by expert users at a very low cost (arguably, any expert in a domain can name more than 5 examples of a named entity).

As an example of this step, consider the word "Letor" (i.e., an entity of dataset type) in the seed term list. All sentences in the containing the word "LETOR" in the corpus, such as *"We performed a systematic set of experiments using the LETOR benchmark collections OHSUMED, TD2004, and TD2003"* are extracted, and provide as examples of the positive classification class. We also extract surrounding sentences in the text to better capture the usage context of the seed entity.

Expansion

As seen in the sentence example provided in the previous section, also `OHSUMED`, `TD2004` and `TD2003` are identified as belonging to the dataset entity type, but since they are not in the seed terms they will be labeled negatively – thus leading to more false negatives. At the same time, the extraction of sentences in the training data that are related to seed terms will cause a shortage of negative examples for training purposes. In order to avoid these problem the *term expansion* and *sentence expansion* strategies were introduced.

Term Expansion

Term Expansion is designed to reduce the number of false negatives in the training sentences and provide more positive examples. *Semantic relatedness* is used: terms which are semantically similar or related to terms in the seed list should be included in the expansion. For example, given the dataset seed term `LETOR` , the expansion should add semantically related terms like `OHSUMED` or `TD2004` which are also benchmarks used in the field of information retrieval. First the *word2vec* model [153] is trained on the whole corpus by learning all uni- and bi-gram word vectors of all terms in the corpus. Then, NLTK entity detection is used to obtain a list of all entities contained in the sentences of the training data and cluster them with respect to their embedding vectors using K-means clustering. Silhouette analysis is used to find the optimal number k of clusters. Finally, clusters that contain at least one of the seed terms are considered to contain entities of the same type (e.g *Dataset*)

Sentence Expansion

Sentence Expansion (SE) strategy is designed to addresses the problem of the over-representation of positive examples and to increase the size and variety of the training set. The goal of this step is to include sentences that are similar in semantics and vocabulary to the original training sentences, and are unlikely to contain instances of the desired type, to serve as informative negative examples for boosting the NER training accuracy. First the *doc2vec* document embeddings [111] is used, to learn vector representations of

the sentences in the corpus. For each sentence in the training data, *doc2vec* is used to discover the most similar sentence which does not contain any known instance of the targeted type (i.e., expanded terms).

Training Data Annotation

After obtaining an expanded set of *seed terms* and *training sentences*, if any of the words in the *seed terms* matches a word in the *training sentences*, the word will be labeled positively. The annotated dataset can be used as an input to train any state-of-the-art supervised NER algorithm

NER Training

For training a new NER, the Stanford NER tagger[2] is used to train a Conditional Random Field (CRF) model. CRF learns the hidden structure of an input sequence by defining a set of feature functions (e.g. word features, current position of the word labels of the nearby word), assigning them weights and transforming them to a probability to detect the output label of a given entity.

Filtering

In this final step, which is also the focus of this work, the trained NER model is used to annotate the whole corpus and consider all the positively annotated terms as candidate terms for the next round of iteration. As the training data for training the NER is noisy, the list of entities extracted by the NER contains many items which are not specifically related to the entity type of interest. Therefore, the goal of this last step is to filter out all terms which are most likely not relevant using the following basic heuristics [147]:

Wordnet + Stopwords (WS) Filtering. Filtering stopwords (e.g. something) and concepts coming from "common" English language (e.g., "dataset", "software") that could be found in Wordnet[3].

Similar Terms (ST) Filtering. Excluding entities that do not appear in the same cluster that contains a seed term - i.e. explained in 4.2.

Pointwise Mutual Information (PMI) Filtering. Filtering entities having a semantic similarity measure (i.e. derived from the number of times two given keywords appear together in a *sentence* in our corpus) lower than a threshhold.

Knowledge Base Lookup (KBL) Filtering. Excluding entities that have a reference in the DBpedia knowledge base (under the assumption that, if they are mentioned in DBpedia, then they are not long-tail domain-specific entities.

Ensemble Majority Vote (EMV. Preserving the entities that are passed through two out of three selected filtering strategies.

[2]https://github.com/dat/stanford-ner

[3]4

Interested readers can refer to [147] for detailed explanation. As those heuristics are rather basic in their nature, we discuss in the next section of filtering can be supported by human feedback.

4.3 Collaborative Crowd Feedback

As outlined in the previous section, a core design feature of TSE-NER is the heuristic filter step in each iteration, which is designed to filter out named entities which are most likely misrecognized (this can easily happen as the used training data is noisy due to the strong reliance on heuristics). While it was shown in [147] that this filter step indeed increases the precision of the overall approach, it does also impact the recall negatively. For example, this could happen by filtering out *true positives*, i.e. entities which have been correctly identified by the newly trained NER extractor but are filtered out by the heuristic. This could for example happen if a domain-specific named entity is part of common English language. More importantly, the heuristic filter often does not reach its full potential by not filtering all *false positives*, i.e. entities which are incorrectly classified as being of the type of interest, and should have been filtered out by the heuristics but were missed. Also, for the expansion phase, the heuristics often miss relevant entities which should be added.

These shortcomings are addressed in this chapter by introducing an additional layer on top of the basic TSE-NER training cycle described in Section 4.2. Instead of treating the algorithm only in isolation, we also consider the surrounding production system and its users (in most cases, this would be a digital library repository with search, browsing, and reading/downloading capabilities). When the production system is set-up, a NER algorithm is trained for each entity type of interest (e.g., datasets, methods, and algorithms for data science) using the TSE-NER workflow until training converges towards stable extraction performance. Then, the resulting trained NER algorithm is applied to all documents in the repository, annotating their full-texts. Users then can interact with the recognized entities, providing feedback.

For this, we introduce novel Coner modules (see Figure 4.1):

1. **Coner Interactive Document Viewer (CIDV)**: Online interactive viewer that renders PDF documents and visualises automatically annotated entities. The CIDV allows users to interact with entities by giving feedback on existing annotations, or adding new typed named entities.
2. **Coner Feedback Analyser (CFA)**: Analyzes the entity type labels for each entity that received human feedback, and also decides which labels should be considered valid and which ones are irrelevant. This feedback is then incorporated into the iterative TSE-NER training.
3. **Coner Document Analyser (CDA)**: In an user experiment like the one presented in this chapter, the CDA selects the documents where user feedback would be most effective (see Section 4.4). In a real-life deployed version of Coner, users

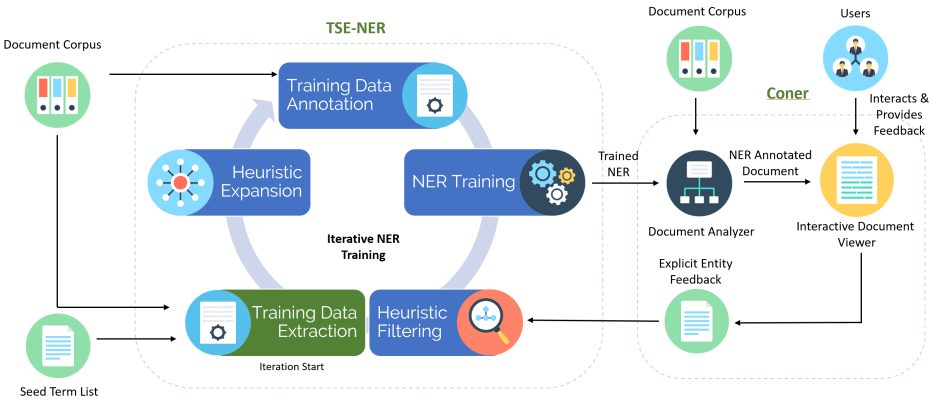


Figure 4.1: Overview of Coner Collaborative NER Pipeline: Human Feedback influences the TSE-NER filter phase, supporting or superseding heuristic decision making

would continuously provide feedback on documents they are currently reading as part of their normal consumption workflow, so no document selection is necessary.

Coner Interactive Document Viewer CIDV

We introduce an *interactive document viewer*, rendering PDF documents and highlighting recognized named entities. The viewer is based on the NII PDFNLT [3, 6], which already included a basic viewer and a sentence annotation tool. One of our design goals for the interactive viewer component was to impose as little cognitive load on the users as possible, thus only very simple feedback mechanisms have been considered. During our proof of concept testing phase, we recruited 10 lab student of graduate or post-graduate level to stress test and give feedback on the viewer. Based on the feedback of these users, we opted for a system design allowing for simple YES/NO relevance feedback for recognized entities. Furthermore, users can add new typed entities by selecting n-grams in the document and assigning an entity type (Figure 4.2). For other users, these manually added entities are also highlighted, and additional user feedback can be provided for them.

Coner Feedback Analyser CFA

The purpose of the feedback analyzer is to aggregate collected user feedback on entities, and decide which new entities to finally add and which entities to label as incorrectly typed. In the current version of the feedback analyzer, this is realized with a simple majority vote on the user feedback.

However, like with any crowd-sourcing task, the feedback analyzer can be further extended to cope with common crowd-sourcing problems like spam, malicious indent, or incompetent users. For example, while for our prototype system maliciousness was not an

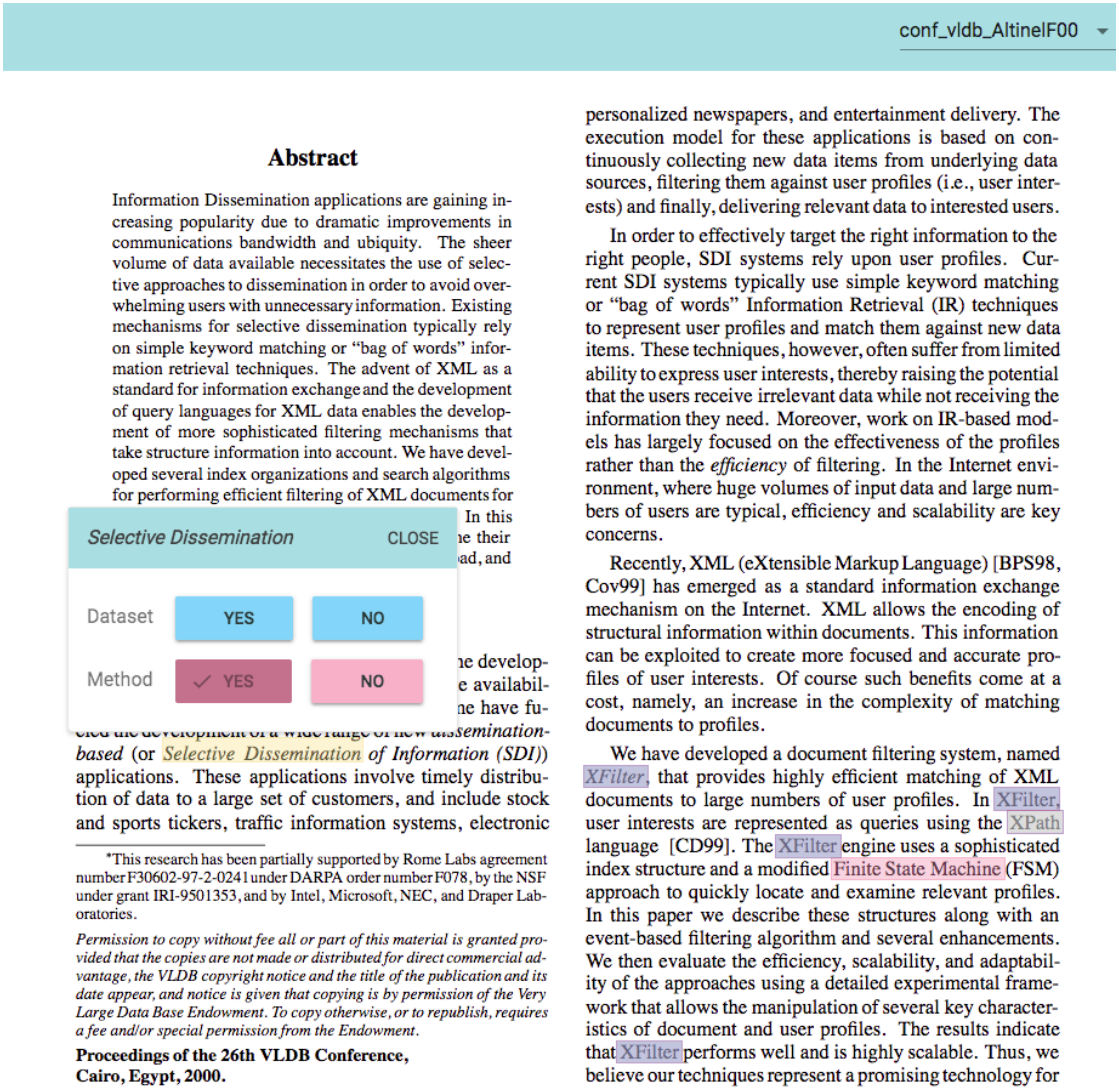


Figure 4.2: Coner Interactive Document Viewer with highlighted entities

issue, we could already see that some users were significantly more reliable than others. This also reflects in their time investment: more reliable users took much longer to provide feedback on a document, while some users provided feedback in a time frame which should not be sufficient for even reading the paragraphs surrounding an entity. Here, more complex user and task models should help to increase the reliability of aggregated user annotations. As a minimalist step towards this, we only consider users who provided feedback on at least 10 entities per publication, and only considered majority votes with at least 3 votes.

As described in Section 4.2 and 4.2, TSE-NER expands and filters the current set of terms every iteration. Coner boosts this process by adding or removing entities from the iterations. Filtering heuristics can be used individually or in an ensemble. Ensemble filtering was shown to have the best, but still limited performance [147]. Coner overwrites the filtering heuristics by ensuring that entities which were labeled by users as irrelevant for a type are always removed during filtering, and entities labeled as relevant are always retained. Similarly, for the heuristic expansion step, we ensure that manually added entities are always included.

4.4 Evaluation

To evaluate the effectiveness of incorporating crowd feedback into the NER training process, we focus on the following two research questions:

RQ1 What are the properties of obtained user feedback? Especially, in how far does human feedback confirm or conflict with TSE-NER heuristics?

RQ2 How does incorporating human feedback into the TSE-NER filtering step improve the overall performance with respect to precision, recall, and F-measures?

To answer these research questions we conducted two user experiments. Similar to [147], we focus on the two entity types *dataset* and *method* in data science publications. We had corpus of 11,589 papers from 10 conferences on data science available (this is the same corpus as used by [147]). We conducted the user interaction with the Coner system in a lab setting, recruiting graduate-level / post-graduate-level volunteers who are knowledgeable in the data science domain.

The first experiment, as described in Section 4.4, focuses on answering RQ1 by asking users to give feedback or add to unfiltered extracted entities (i.e., on the output of TSE-NER using expansion but no filtering heuristics). By comparing crowd-based filtering to the different filter heuristics, we can obtain insights into their relative performance.

As we only had a limited number of volunteers available for this evaluation, we selected papers from our corpus using the Coner Document Analyser CDA for which the expected impact of additional annotations is representative for the whole collection. CDA is only used for experimental evaluations, and we define representatives of a publication as being published at a higher-level conference, having average length, high citation counts, and an average number of distinct recognized typed entities in their full texts.

Furthermore, for the second experiment in section 4.4, instead of relying on our users to decide themselves on which entity to provide feedback, we actively steer this process towards entities for which human feedback would have a significant expected impact. In particular, we focus on entities which were classified as both *dataset* and *method*. This happened quite often (i.e. 22% of all the detected entities in the whole corpus), and in nearly all cases, at least one classification is incorrect. We divert the decision which of the two types (if any) is correct for the entity to our system's users.

Experiment 1: Human Feedback on Unfiltered Entities

In this section, we look into the properties of user feedback itself, and also evaluate how it conflicts or supports TSE-NER heuristics.

Documents and Evaluators: Ten papers were selected from multiple conferences of interest using the Document Analyzer. We selected from the following conferences: The Web Conference (3 papers), ACL (3 papers), ICWSM (2 papers) and VLDB (2 papers). The selected documents contain overall 255 distinct recognised *dataset* entities before filtering, and 85 distinct recognized *method* entities before filtering. The average number of times each selected paper has been cited is 581. The 10 human evaluators are randomly and uniformly assigned to the documents such that each document is processed by at least 3 evaluators. We obtained this minimum threshold of three users' feedback on the recognition correctness on 271 *dataset* entities (94.8%) and 158 *method* entities (94.0%). Note that users could add new entities, increasing the number of distinct entities. The evaluators showed quite varying task completion times for giving feedback on all entities contained in a document, with an average of 7:57 minutes to provide feedback for a single document, while the fastest evaluator only needed 3:14 minutes and the slowest 19:38 minutes.

Entities and Agreement: The evaluators were not forced to rate all occurrences of recognized entities (the assignment was: "provide feedback on the recognized entities as you see fit."). The average percentage of recognized entities (highlighted in the Coner Viewer) each evaluator gave feedback on is 65.9%. There were no discernible differences between *dataset* and *method* entities. After the experiment we interviewed the evaluators on their reasons for skipping feedback: First, ambiguous meanings of the same entities annotated in different sections and contexts caused doubt about type relevance (e.g. the named entity `Microsoft` can reference a dataset created by Micrsoft or the actual company itself). Second, some bigram or trigram *method* entities were recognized with additional useless trailing words (e.g. *question taggings have*), therefore also not receiving feedback from some evaluators.

Table 4.1 compares the percentage of *dataset* and *method* entities that where considered correct by the TSE-NER classifier (i.e. without the filtering step) or manually added by an evaluator, but judged as incorrect by the majority of evaluators. The false positive rates in Table 4.1 indeed show the effectiveness of collaborative feedback on TSE-NER. Interestingly, not all of the named entities added by users were rated as relevant for their intended type; for user added entities, we observe a false positives rate of 25.9% for *dataset* and 11.7% for *method*. This means that it is crucial to also receive user feedback

	Dataset (FP%)	Method (FP%)
User added	25.9%	11.7%
NER extracted	94.3%	57.9%
Total	**80.4%**	**27.4%**

Table 4.1: Comparison of false positive rates, resulting from majority vote on relevance of unfiltered extracted entities for both user added and NER extracted entities

	PMI	WS	ST	KBL	EMV	CB
Dataset	9.0%	86.9%	34.4%	90.7%	35.0%	19.5%
Method	9.4%	73.7%	69.0%	81.2%	41.6%	52.2%

Table 4.2: Comparison of entity retention rate between Coner and TSE-NER filter techniques (315 entities for *dataset* and 198 entities for *method*. Filtering acronyms: Pointwise Mutual Information (PMI), Wordnet + Stopwords (WS), Similar Terms (ST), Knowledge Base Look-up (KBL), Ensemble Majority Vote (EMV), Filtering Coner Boost (FCB): EMV + Coner Human Filtering

from evaluators on entities other users added to ensure the quality of human feedback. Evaluators differ in skill, expertise, and also effort they put into feedback, which clearly influences their decision making.

We calculated the average Cohen's Kappa between the 10 evaluators for each entity type. On average, Cohen's Kappa for *dataset* entities is 0.51, while for *method* entities it is 0.63.

Comparison Filtering Techniques: Coner vs TSE-NER
Table 4.2 compares the performance of Coner human feedback filtering and different filtering heuristic setups for TSE-NER in terms of retention rate; the percentage of unfiltered extracted entities kept by each filter. The different filtering techniques were performed on the complete set of entities that received feedback from at least three evaluators in the 10 selected papers; 315 *dataset* and 198 *method* entities. As illustrated in Table 4.2, the Coner Boost (FCB) filtering technique described in this chapter is more strict than Ensemble Majority Vote originally used by TSE-NER for the *dataset* type, but less strict for the *method* type. This can be explained by the larger percentage of user added named entities for the *method* type compared to the *dataset* type, with user added named entities having a much lower average false positive rate compared to NER extracted entities (Table 4.1).

To get a better insight into the filtering performances, we compared the false positives rate for each filtering technique with regards to the set of entities determined to be relevant by human evaluators (Table 4.3); if an entity is kept by a filter for a type, but was voted as irrelevant for a type by the majority of evaluators, then it is considered a false positive instance. For most of the TSE-NER filtering setups the average false positives rate for both facets is above 50% (only PMI has a lower false positive rate, because it is much more selective in its retention of entities). This means there are a significant number of entities that were recognised as irrelevant for a type by human

	PMI	WS	ST	KBL	EMV	FCB
Dataset	38.7%	73.9%	79.7%	79.4%	76.7%	8.8%
Method	25.0%	28.2%	40.3%	37.7%	37.7%	3.9%

Table 4.3: Percentage of false positives in the remaining filtered entity sets of TSE-NER filtered heuristics compared to Coner human filtered entities. Filtering acronyms same as Table 4.2

	PMI	WS	ST	KBL	EMV	FCB
Dataset	76.2%	3.8%	70.0%	20.0%	65.0%	0.0%
Method	88.2%	4.6%	30.9%	15.1%	56.6%	1.3%

Table 4.4: Percentage of false negatives in the remaining filtered entity sets of TSE-NER filtered heuristics with regards to Coner filtered entities.

judgement, but TSE-NER heuristic filtering was unable to do so.

We also considered the false negatives which were excluded by the filtering techniques but were labelled as relevant by majority of evaluators (Table 4.4). The PMI filtering as explained in [147] achieved the highest precision among the TSE-NER filtering techniques in their evaluation. Table 4.4 clearly indicates a major shortcoming of the PMI filtering heuristic; it filters out on average 82.2% of Coner viewer entities that were rated as true positives by Coner human feedback. Even for the EMV filtering heuristic, which is regarded as most effective in terms of F-Score by [147], the average false negatives rate is 57.8%. Also, in Table 4.2 we see that KBL has the highest average retention rate of named entities, which also translates in a high false positive rate and lower false negatives rate.

Finally, Table 4.3 and Table 4.4 demonstrate that the FCB filtering approach results in the lowest false positives and false negatives rates compared to Coner human filtering; this is good for the quality of filtered entities, because more relevant named entities overlap with the Coner human filtering (regarded as true positives), but it also means it difficult to scale this approach with a significantly larger number of named entities.

Qualitative Entity Inspection: When there is a user consensus, Coner removes or adds entities to the TSE-NER expansion and filter phases, effectively overwriting the heuristics. We manually inspected some of these entities to obtain an intuition on what entities the TSE-NER heuristics usually fail at. Table 4.5 shows some randoms sample entities which have been consensually labeled as wrong with respect to the recognized type, while table 4.6 shows entities which are labeled as correct. Table 4.7 shows some samples which failed to obtain user concensus and obtained a mix of positive and negative labels.

For example users seem to be uncertain and fail to reach consensus when entities are related to a type but are too generic, e.g. `signed networks, news article, news feed, data base, etc.` for *dataset* and `algorithm, decision rule and used search algorithm` for *method*. This could be explained by a difference in domain expertise or interpretation of what belongs

Dataset	digg interfaces, logistic regression, acyclic subgraph
Method	digg, flickr, wikipedia, dynamic programming, system description

Table 4.5: *Dataset* and *Method* annotated entities annotated as incorrect

Dataset	digg, flickr, wikipedia, datasets,
Method	hybrid multimodal method, similarity search, reinforcement learning, logistic regression, acyclic subgraph

Table 4.6: *Dataset* and *Method* annotated entities annotated as correct

Dataset	signed networks, slash, news article, news feed
Method	10-foldcross validation, algorithm, decision rule, used search algorithm, vldb, web services

Table 4.7: Sample of *Dataset* and *Method* annotated without clear user consensus

to a certain type between evaluators.

This shows that even for humans, reliably typing entities is hard as there is quite some room for subjective interpretation.

Also, in during our inspection, we encountered frequently entities which are classified both as *method* and *dataset* by TSE-NER like `digg, flickr, wikipedia, logistic regression, acyclic subgraph`. Most of these double classifications are wrong, and we will further investigate this double classification phenomenon in Section 4.4.

Experiment 2: NER Performance

We picked 28 papers from 4 conferences in our document corpus, similarly to our document selection described in section 4.4; 13 papers from VLDB, 9 papers from The Web Conference, 4 from SIGIR and 2 from ICWSM.

We recruited 15 graduate-level/post-graduate-level volunteers and instructed them to focus their efforts on judging entities recognized in these papers. However, for this experiment we want to make sure that user feedback is as effective as possible to use our human annotators time efficiently. As a heuristic we focus on entities which have been double-classified as both *dataset* and *method*, and thus one of the types is nearly guaranteed to be wrong. As mentioned before, double classifications between *dataset* and *method* are quite common. This can be explained by the relative similarity of these

	Dataset (P/R/F)	Method (P/R/F)
TSE-NER	0.66/0.60/0.63	0.56/**0.21**/0.30
Coner	**0.70**/0.60/**0.65**	**0.59**/0.20/0.30

Table 4.8: Comparison of performance of *TSE-NER* and *Coner* in terms of Precison/Recall/F-Score for two type of doubly filtered entities: *Dataset* and *Method*

two types: both types appear in similar contexts and/or sentence structures, and are much closer to each other than typical entities types considered in NER like *location* and *person*. Thus, distinguishing between *dataset* and *method* can be considered a very hard task for an automatic classifier. Cases like these is when user feedback is the most valuable.

In order to measure the effect of human feedback into the TSE-NER filtering, we repeat the experiments described in [147] and use the same test set, measuring the F-Score, precision, and recall with and without the Coner feedback. We used the output of the experiment and the TSE-NER to train the NER model. For training we used 71,292 and 103,568 (i.e, *dataset* and *method* entity type) sentences for TSE-NER and 25,819 and 53,200 (i.e, *dataset* and *method* entity type) sentences for Coner and employed the SE strategy. For testing, 3149 sentences were used for dataset and 1097 sentences for method entity type.

Table 4.8 compares the performance of TSE-NER with and without Coner feedback focused on double-classified entities in terms of precision, recall and F-Score. As shown in Table 4.8 there is an increase in precision for both *dataset* and *method* type classifiers when incorporating user feedback with Coner, while recall and f-score remains stable. Naturally, providing feedback on recognized entities as part of the filter step cannot increase recall, but only affect precision by removing *false positives*. Overall, the test dataset covered 555 unique entities, and we obtained user feedback on 29 unique entities of the test set. Nonetheless, this shows that by focusing user feedback on parts which are in doubt, like the double-classified entities, even a smaller number of user feedback can make a difference, i.e. by obtaining feedback on only 0.05% of the entities in the test set we could increase the precision by 4%. This significant increase in precision is mainly due to the fact that user feedback improves the quality of the input data for each training iteration of TSE-NER, thus the effect of each feedback is greatly magnified. In a scenario where Coner is constantly running in the background, we expect notable increases both for precision and recall (due to allowing users to suggest new entities).

4.5 Related Work

A considerable amount of literature published in recent years addressed the *deep analysis* of text such as topic modelling, domain-specific entity extraction, etc. Common approaches for *deep analysis* of publications rely on techniques such as dictionary-based [198], rule-based [55], machine-learning [192] or hybrid (combination of rule based and machine learning) [205] techniques. Despite its high accuracy, a major drawback of dictionary-based approaches is that they require an exhaustive dictionary of domain

terms. These dictionaries are often too expensive to create for less relevant domain-specific entity types. The same holds for rule-based techniques, which rely on formal languages to express rules and require comprehensive domain knowledge and time to create. The lack of large collections of labelled training data and the high cost of data annotation for a given domain is one of the main issues of machine learning approaches. Many attempts have been made to reduce annotation costs such as bootstrapping [200] and entity set expansion [23, 101] which rely only on a set of seed terms provided by the domain expert. Unfortunately, this reliance on weak supervision just providing seed terms limited also the maximal achievable performance with respect to precision, recall, and F-scores.

Active learning is another technique that has been proposed in the past few years, asking users to annotate a small part of a text for various natural language processing approaches [190, 211, 72] or generating patterns used to recognize entities [132]. With active learning, the unlabelled instances are chosen intelligently by the algorithm (e.g. least confidence, smallest margin, informativeness, etc) for annotation. Furthermore, combining an active learning approach with uncertainty sampling as retraining annotation selection method has been widely researched [216, 115, 228, 227, 189].

The proposed approach in this chapter is inspired by active learning techniques [190, 211, 72] but relies on training NER algorithms for long-tail entities in a distantly-supervised fashion which incrementally incorporates human feedback on the relevance of extracted entities with high expected information gain into the training cycle. In addition, in contrast to [72] where the authors just present bibliographic sentences to Amazon Mechanical Turk annotators for labelling, our work focuses on the annotation of long-tail entities which relies on the occurrence context for easier annotation.
We incorporate collaborative user feedback on type relevance of classified entities and annotation of new entities to continuously support the sentence expansion and entity filtering steps of the iterative TSE-NER algorithm [147]. Newly annotated relevant domain specific entities are added to the seed set in the expansion step, to fetch additional relevant training sentences and terms to increase the number of true positive occurrences in the training data. Furthermore, we allow to filter out irrelevant entities in the filtering step, to reduce the number of false positives detected by the noisy NER.

4.6 Conclusion and Future Work

In this chapter, we introduced Coner, a collaborative approach for Long-tail Entity Recognition in scientific publications. Coner extends the TSE-NER technique for iterative training of NER algorithms using distant supervision [147]. To keep the training costs low, TSE-NER relied on heuristics to steer the training process (i.e., by expanding and filtering entity sets), requiring only on a small seed set of known named entities of the desired type as manual input. Unfortunately, this reliance on automatic heuristic expansion and filtering limited also the maximal achievable performance with respect to precision, recall, and F-Score.

We approached this problem with a unique solution: we considered the synergy between NER training and the productive system it is employed in, including the respective user base. In particular, Coner allows us to mostly automatically train a NER algorithm at very low cost, and then exploit the daily user interaction (like in a digital library system) for continuously improving the algorithm's performance, requiring only simple and intuitive feedback actions from the users. This is realized with an *interactive viewer component*, which allows users to elicit feedback on the correctness of recognized entities unobtrusively while reading the document.

In this work, we focused on augmenting the filter step of TSE-NER by incorporating user feedback into the NER training process. Our lab experiments showed that 94.3% for *Dataset* and 57.9% for *Method* of entities detected by partial TSE-NER without heuristic filtering were indeed false positives. We observed that by using different filtering heuristics, we could reduce the number of false positives up to 38.7% for *Dataset* and 25% for *Method* (i.e., using the PMI filtering heuristic) which also results in higher false negatives rate as shown in Table 4.4. In order to reduce the number of false positives as well as false negatives we proposed incorporating user feedback into filtering which resulted in the lowest false positives (i.e., 8.8% for *Dataset* and 3.9% for *Method*) and false negatives (i.e., 0.0% for *Dataset* and 1.3% for *Method*). Furthermore, we showed that by obtaining feedback on only 0.05% of the entities in the test set (and others outside the set), we could increase the precision by 4% while keeping recall and f-score stable.

For future work, we can leverage Coner's full potential by integrating it into an existing production system, like a large scale digital library. In this case, we can receive continuous feedback from the system's users on a number of papers magnitudes bigger than our private lab experiment conducted so far and improve the performance of the NER techniques over time. Likely, user feedback techniques usable for term expansion will require a heavier toll, and thus need further investigation. To a certain extent, this could be offset using appropriate *incentivation* techniques: by motivating the user to be willing to contribute feedback (for example by means of gamification), even more elaborate feedback mechanisms could be employed without degrading user satisfaction. However, as with all systems relying on crowd-sourcing or explicit user feedback, *fraud* and *vandalism* become a central concern. If Coner is to be used with real-life users outside of a lab setting, such issues need to be addressed by for example user reputation management [45] or different voting consensus techniques [57]. While the techniques introduced in Chapters 2, 3 and 4 have indeed shown to reduce the cost of training and improve the overall performance of Long-tail Entity Recognizer, they are typically limited by the availability of the words and sentences in the semantic space (Chapter 3) and the availability of continuous feedback from users (Chapter 4). This leads us to our next chapter where we focus on generating new text not existing in the corpus, thus largely expanding the training data in a cost efficient manner.

Chapter 5

Training Data Augmentation Using Deep Generative Models

This chapter investigates RQ4 and introduces another technique for augmenting training data for Long-tail Entity Recognition. To further our understanding of how to augment training data for the extraction and typing of long-tail entities in other sources than scientific publication, we look into User generated content (UGC). We focused on user-generated phrases related to type Adverse Drug Reaction (ADR) mentioned in User Generated Content (UGC) such as Twitter and Reddit. Social media provides a timely yet challenging data source for adverse drug reaction (ADR) detection. Existing dictionary-based, semi-supervised learning approaches are intrinsically limited by the coverage and maintainability of laymen health vocabularies. In this chapter, we introduce a data augmentation approach that leverages variational autoencoders to learn high-quality data distributions from a large unlabeled dataset, and subsequently, to automatically generate a large labeled training set from a small set of labeled samples. This allows for efficient social-media ADR detection with low training and re-training costs to adapt to the changes and emergence of informal medical laymen terms. An extensive evaluation performed on Twitter, and Reddit data shows that our approach matches the performance of fully-supervised approaches while requiring only 25% of the training data. The contribution of this chapter is published in [148].

5.1 Introduction

Adverse Drug Reactions (ADRs) is the fourth leading cause of death in the United States [28]. ADR detection is, therefore, a critical element of drug safety. Studies have shown that clinical trials are not able to fully characterize drugs' adverse effects [83, 28, 5]. Traditional techniques of post-market ADR mainly rely on voluntary and mandatory reporting of ADRs by patients and health providers, but they suffer from delays in reporting, under-reporting, or data incompleteness [183].

Social media is becoming a preferred channel for millions of users and patients to share, discuss, and seek health information [86]; such user-generated content can, therefore, provide valuable insights for monitoring Adverse Drug Reactions (ADRs) from an additional point of view [114, 184, 9]. ADR detection from social media is, however challenging, as online users report ADRs using a different language style and terminology that largely depend on the user's medical proficiency, as well as the type of online medium (e.g., health forums vs micro-post social networks). In particular, laymen often use diverse dialects [99] when describing medical concepts, and make abundant use of informal terminology.

Existing approaches for detecting terms in informal medical language mainly rely on semi-manually generated dictionaries (e.g. laymen health dictionaries) [221], or supervised machine-learning-based sequence classifiers [92, 34]. Due to the language dynamicity in online and offline communication [102, 219], there is a constant emergence of new informal medical terms. This results in a lack of coverage and maintainability of laymen health vocabularies. While showing superior performance, machine learning approaches often need to be trained for specific Web communities and platforms due to differences in the underlying language models; this results in high costs for manual annotation of training data, which for many domains is only sparsely available [50].

More recently, researchers have started to investigate techniques for expanding the size of manually created training data. Often, sentence similarity implemented with embedding techniques [153, 111] is used to discover similar sentences, and then annotations are automatically propagated to those sentences [146]. While these techniques have indeed shown to reduce the cost of training, they are typically limited by *availability of the existing data* as the reliability of annotation propagation suffers when sentences are not similar enough. Therefore, in this chapter we focus on how to automatically *generate* high quality training data for Adverse Drug Reaction detection with minimal human supervision and costs.

Original Contribution. In this chapter, we propose to generate artificial sentences closely mimicking existing training data; such artificial sentences are then annotated automatically via label propagation. This contrasts existing approaches expanding manually created training data set by discovering additional existing sentences in a dataset.

We build our method upon variational autoencoders (VAE), a deep probabilistic neural model which learns latent text features and their distributions effectively. In contrast to other approaches using variational autoencoders for text generation, we modify the mechanism for generating new artificial samples such that we obtain samples structurally

and semantically similar to a specific subset of the original data. This allows us to generate sentences similar to those in the pre-existing human-labeled ADR training usable for classifier training set by taking advantage of the implicit semantics contained in the larger unlabeled dataset.

We evaluate the proposed method on a standard Twitter dataset and on a large new dataset for the Reddit platform we created with the help of expert annotators. The dataset is available on the companion Website [37]. Results show that our approach achieves superior or comparable performance with significantly less training data (*reduced by 75%*) than state-of-the-art training methods. This approach can be also used for extracting long-tail entities from scientific publications. The effectiveness of this approach on scientific publication needs to be investigated in future work.

5.2 Related Work

The terminology adopted in most social communities makes heavy use of slang or indirect descriptions, which is often lacking with respect to grammar and orthography; in addition, it is also constantly evolving and differs between communities. This makes the use of established techniques relying on expert-curated dictionaries [113, 196] of consumer health vocabulary or fully-supervised machine learning-based classifiers [92, 34] expensive, and in many cases even prohibits their use. While techniques to lessen the costs of training like distant supervision [155] or bootstrapping [200, 17] can provide some support, their performance has been shown to be limited.

Some recent work has started to address the issues of size and cost of ADR training data [114, 35, 162]. Lee et al. [114] explores different types of unlabeled data and a small training set to generate phrase embeddings, so to *classify* the tweets that indicate adverse drug event in a semi-supervised way. In contrast, our work focuses on detecting the *actual ADR span* in the text of the user generated posts rather than just classifying the whole post as containing an ADR mention. Nikfarjam et al. [162] and Cocos et al. [35] augment traditional supervised methods with additional features such as pre-trained word representation vectors, to improve performance and to be less dependent on large training sets. The resulting BLSTM-RNN [35] technique, which achieves state-of-the-art performance, is also evaluated in our experiments (see Section 5.5). Rather than adding new features or proposing new ADR detector models, our work focuses on the generation of new labeled data samples from small annotated training sample using deep probabilistic models.

Different from the above approaches, embedding based methods [111, 153] learn vector representations of words or paragraphs to capture semantic relationships among words. Such methods are, therefore, useful to find sentences similar to the labeled training data, thereby expanding the size of the training data. Embedding based methods, however are limited by the existing sentences available in a given corpus. In contrast, our approach is capable of generating new sentences not existing in the corpus, thus largely expanding the training data. Our approach for generating additional labeled training data is inspired by [21], where VAEs are used to learn a generative model of text for sentence generation.

Bowman et al. [21], however, only tackles the general problem of sentence generation. To the best of our knowledge, our work is the first that investigate VAEs as a tool for training data expansion, so as to enhance machine learning performance with limited amount of labeled data.

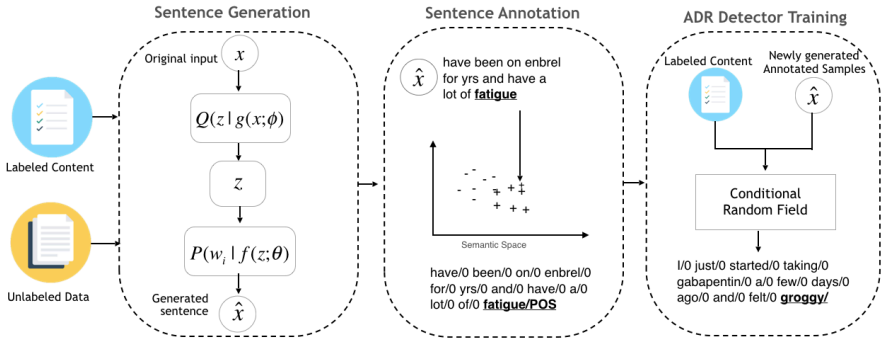


Figure 5.1: Overview of the proposed Training Data Augmentation Approach for Adverse Drug Reaction Detection

5.3 Adverse Drug Reaction Detection in User Generated Content

Approach Overview. Figure 5.1 presents an overview of our proposed approach. Given a list of drug names, a corpus $UGC = \{ugc_1, .., ugc_n\}$ of health-related User Generated Content (UGC) that mention one or more drugs of interest, and a subset $LC \subset UGC$ of UGCs labeled with ADR mentions, the *Sentence Generation* step (Sections 5.3) creates a set SC of newly generated sentences that are similar to the ones in LC. The size of LC is usually highly limited, thus Sentence Generation is important to expand the labeled data for better training the ADR detector. In LC, terms related to ADR (e.g. "no appetite") are considered positive examples ($POSTerms$), while all the other terms (e.g. "aspirin", "again"), excluding English stop words, are considered negative examples ($NEGTerms$). A Variational Auto Encoder (VAE) is first trained on UGC data to learn the data distribution of the dataset, and then provided with LC sentences as input to generate the output set SC. The *Sentence Annotation* step (Section 5.3) then propagates the label from LC to the terms of sentences in SC. This is achieved by labelling the set $UTerms = \{ut_1, .., ut_n\}$ of terms in the newly generated sentence SC that are semantically more similar to $POSTerms$ than to $NEGTerms$ in LC. Finally, the labelled sets LC and SC are combined in the *ADR Detector Training* step (Section 5.3) to train an ADR detector.

Sentence Generation

Our method for data generation relies on learning sentence distributions from a large text corpus, which can then be used to generate posts SC semantically similar to a

Table 5.1: Three samples generated using VAE for a given input sentence.

Input	**my dr switched from celexa to paxil and paxil made me feel sick**
Sample 1	my doctor put me on cymbalta and cymbalta can help me function
Sample 2	took my fluoxetine and it was a bit spaced out of my brain
Sample 3	yeah have to take topamax and it helps me but still feel fuzzy headed to a bit
Input	***bruh this vyvanse putting me to sleep I needa take a break***
Sample 1	took my vyvanse today and my head is spinning
Sample 2	vyvanse makes me feel like a zombie
Sample 3	vyvanse and addy have a cup of coffee
Input	***I was on Prozac for months but it made my emotions so suppressed I stopped taking them***
Sample 1	I was on venlafaxine for anxiety and depression but it stopped working
Sample 2	I was on effexor for about 3 months and then switched to venlafaxine
Sample 3	was on latuda for a while but it didn't help me

given set of existing labeled content LC. Let $\mathbf{x} \in \mathbb{R}^{|V|}$ ($\mathbf{x} \in UGC$) be the bag-of-words (multi-hot) representation of a user-generated content, where V is the global vocabulary, and $\mathbf{w}_i \in \mathbb{R}^{|V|}$ be the one-hot representation of the word at position i in the sentence represented by $\mathbf{x}$. Our goal is to learn $P(\hat{\mathbf{x}}|\mathbf{x})$, where the probability of a newly generated content $\hat{\mathbf{x}}$ serves as a proxy of the semantic similarity between $\hat{\mathbf{x}}$ and the original labeled content $\mathbf{x}$. Note that we will use the full set of user generated content UGC to learn the data distribution, while only the labeled subset LC is used to generate new sentences.

To obtain this conditional distribution, we adopt the deep generative modeling approach [106, 149], which was originally proposed to generate data instances similar to those already in a given dataset. Here, data is embedded into a latent space which is modelled by conditional distributions, and samples from this distributions can be decoded into new artificial data instances. In contrast to shallow models such as Skip-Gram [153] which also embeds into latent spaces, deep generative models have been shown to capture the implicit semantics and structure of the underlying data more effectively. However, existing deep generative models are not designed for generating class-specific data instances. Therefore, our goal is to extend existing deep generative models such that we can choose to only generate samples of a chosen subclass (e.g., resembling just labeled data). For example, Table 5.1 shows 3 artificial samples generated for 2 human-written training data sentences.

To do so, we build our method upon variational autoencoder (VAE), a representative deep generative model capable of learning high-quality representations of data structures. Given a set of sentences, VAE aims at learning a likelihood function $P_\theta(\hat{\mathbf{x}}|\mathbf{z})$ that, when used together with a standard Gaussian prior of $\mathbf{z}$, can generate new data instances $\hat{\mathbf{x}}$ that are similar to existing ones. Here $\mathbf{z}$ is the latent feature vector that captures the underlying data structure of the existing dataset. To handle the complex relationship between the latent feature and textual content, the likelihood function is parameterized by deep neural networks.

Variational Autoencoder. VAE encompasses a generative model, which describes the generative process for new data instances $\hat{\mathbf{x}}$ given $\mathbf{z}$ sampled from the Gaussian prior and transformed through a deep neural network.

- For each user-generated sentence $\mathbf{x}$

- Draw a latent feature vector $\mathbf{z} \sim P(\mathbf{z})$ where $P(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$ is the standard Gaussian distribution.
- For the i^{th} term in the sentence,
  - Draw $\mathbf{w}_i \sim P(\mathbf{w}_i|f(\mathbf{z};\theta))$

where $f(\mathbf{z};\theta)$ is the neural network whose weights are shared for all sentences. The conditional probability over words, i.e., $P(\mathbf{w}_i|f(\mathbf{z};\theta))$ is modeled by a multinomial logistic regression:

$$P(w_i|f(\mathbf{z};\theta)) = \frac{exp(\mathbf{w}_i^\intercal f(\mathbf{z};\theta))}{\sum_{j=1}^{|V|} exp(\mathbf{w}_j^\intercal f(\mathbf{z};\theta))}$$

The parameters of the neural network, i.e., θ, are learned by maximizing the the log likelihood of the observed sentence $\mathbf{x}$. This is non-trivial due to the intractability of the integral over the latent feature vector $\mathbf{z}$. VAE adopts a variational approach to optimise for the lower bound of the log-likelihood:

$$\mathcal{L} = \mathbb{E}_{Q(\mathbf{z}|g(\mathbf{x};\phi)}[\sum_{i=1}^{|\mathbf{x}|} \log P(\mathbf{w}_i|f(\mathbf{z};\theta)] - D_{KL}[Q(\mathbf{z}|g(\mathbf{x};\phi)\|P(\mathbf{z})]$$

This is generally known as the evidence lower bound (ELBO) [16]. In such an ELBO, $\mathbb{E}(\cdot)$ is the expectation and $D_{KL}[\cdot\|\cdot]$ is the KL-divergence between two distributions; $Q(\mathbf{z}|g(\mathbf{x};\phi))$ is a Gaussian distribution $\mathcal{N}(\bar{}, \text{diag}(\text{œ}^2))$ that is again parameterized by a deep neural network: the two parameters of the Gaussian distribution, i.e., $\bar{}$ and œ are both the output of the neural network $g(\mathbf{x};\phi)$.

New Content Generation. Once a VAE is trained on all user-generated content UGC, we take the existing human-annotated content LC (annotated with ADR mentions) as the input for VAE to generate new sentence SC. The generation is performed by making use of the two conditional distributions learned before, i.e., $Q(\mathbf{z}|g(\mathbf{x};\phi))$ and $P(\mathbf{w}_i|f(\mathbf{z};\theta))$. When used together, these distributions form the conditional distribution we are interested for generating new content:

$$P(\hat{\mathbf{x}}|\mathbf{x}) = \int \prod_{i=1}^{|\hat{\mathbf{x}}|} P(\mathbf{w}_i|f(\mathbf{z};\theta))\ Q(\mathbf{z}|g(\mathbf{x};\phi))d\mathbf{z}$$

Content generation can then be performed via sampling from the above distribution. To generate new sentences, we take each sentence from the labeled set LC, and sample a pre-defined number (k) of latent feature vectors $\mathbf{z}_{j=1}^k$ from $Q(\mathbf{z}|g(\mathbf{x};\phi))$. For each sampled $\mathbf{z}_j$, we use it as an input for $P(\mathbf{w}_i|f(\mathbf{z};\theta))$ to generate a sequence of words as the new sentence.

Sentence Annotation

After generating new samples SC similar to LC, the next step is to automatically annotate the terms in the newly generated sentences with ADR mentions such that it can be used to train a sequence-labeling model. In its basic version, we can only rely on the terms in the $POSTerms$ as positive examples of ADRs. However, we will heuristically expand this term set with additional positive examples found in the SC, thus improving the recall of the ADR detector.

In this work we rely on measuring and aggregating the semantic relatedness SR between a term ut_i and all the terms in $POSTerms$ as well as $NEGTerms$. In general, terms which are semantically related to terms in the $POSTerms$ should be considered as positive example. For example, having the terms `fever` and `no appetite` as positive examples, the new terms `weakness` or `body aches` could also be added to $POSTerms$ (because they are considered semantically related due to frequent co-occurrence, following the distributional hypothesis [84]), while `wheelchair` shall be added to $NEGTerms$. To this end, we use the popular *word2vec* implementation of skip-n-gram word embeddings [153]. We define the semantic relatedness $SR_{pos}(ut_i, POSTerms)$ for a term ut_i and the $POSTerms$ as well as $SR_{neg}(ut_i, NEGTerms)$ as follows:

$$SR_{pos}(ut_i) = \frac{\sum_{pterm \in POSTerms} SR_{pos}(ut_i, pterm)}{|POSTerms|}$$

$$SR_{neg}(ut_i) = \frac{\sum_{nterm \in NEGTerms} SR_{neg}(ut_i, nterm)}{|NEGTerms|}$$

Some terms are semantically related to both $POSTerms$ and $NEGTerms$; for instance, terms such as `drugs`, `pills`, and `pharmacy` have a very close SR_{pos} and SR_{neg}. In order to avoid noisy terms which have an overlap in positive and negative semantics, we only annotate a term as positive if it appears in the $POSTerms$; *or* if the semantic relatedness between ut_i and $POSTerms$ is higher than the semantic relatedness between ut_i and $NEGTerms$, and if the distance between SR_{pos} and SR_{neg} is higher than a given threshold *(th)*.

ADR Detector Training

The labeled training data generated in the previous step can then be used to train any kind of supervised sequence tagger for ADRs. Conditional Random Field (CRF) has shown to be an effective technique on different NER tasks [110].We used the popular Conditional Random Field (CRF) sequence model[1] trained using the features listed in Table 5.2. Finally, the trained ADR detector can be used to detect the ADR mentions in our desired user generated content.

[1] `https://github.com/dat/stanford-ner`. Details on the selected features: `https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/ie/NERFeatureFactory.html`.

Table 5.2: CRF training parameters.

useNGrams=true	normalize=true
noMidNGrams=true	useOccurrencePatterns=true
usePrev=trueuseNext=true	useLastRealWord=true
useLemmas=true	useNextRealWord=true
maxLeft=1	lowercaseNGrams=true

Table 5.3: Dataset statistics. **LC**: labeled training set, **UGC**: unlabelled set. Number of sentences, words, and **unique** ADRs.

	LC(Training)			**LC**(Testing)			**UGC**	
Dataset	**Sentences**	**Words**	**ADRs**	**Sentences**	**Words**	**ADRs**	**Sentences**	**Words**
Twitter	693	6557	379	292	2601	154	146K	2.16M
Reddit	7506	133K	543	1820	31708	195	274K	3.65M

5.4 Evaluation

Experimental Settings

We evaluate the performance of our approach using precision, recall and f-score via approximate matching [202]. The focus of our evaluation is on the variation of performance at different fractions of training data and the number of newly generated samples to demonstrate the effectiveness of our proposed approach in reducing costs of manual annotation for training.

Datasets

Experiments are performed on two datasets targeting different Web platforms. We used the publicly available Twitter dataset from *PSB 2016 Social Media Shared Task* for ADR detection [2]. Next, to evaluate our approach on richer textual forum data, we manually created an annotated corpus of Reddit medical subreddits with the help of medical experts. The aforementioned datasets contain only labeled data, but our approach requires in addition a larger corpus of unlabeled data from the same source. We therefore expanded each datasets with new posts, crawled respectively from Twitter and Reddit, that contain at least of one of the drug names contained in a common vocabulary [3]. The properties of each dataset are described in Table 5.3.

Twitter. The *PSB 2016 Social Media Shared Task* Twitter dataset (i.e. collected as explained in [162]) is a widely used manually annotated training data for ADR detection. The original dataset contained a total of 2,000 tweet IDs[4]; at the time of this study we

[2] http://diego.asu.edu/psb2016/task2data.html

[3] http://diego.asu.edu/downloads/publications/ADRMine/drug_names.txt

[4] Due to Twitter's search APIs license, only tweet ids were released

were able to retrieve text from only 643 tweets, which we acknowledge might have an effect on the performance of the trained models.

Reddit Data. Reddit is a discussion website where users share and discuss problems/ideas about different topics. Reddit also contains subreddits such as `AskDocs`[5], `DiagnoseMe` [6], or `Bipolar` [7] where users share information about their health-related issues. To create a labeled training data set, we used the set of drug names mentioned above to collect 1,626 Reddit posts containing at least one drug names. We then recruited a medical doctor to annotate the ADRs (mentions of adverse drug reactions) in the collected posts following the annotation guidelines suggested in [98], which specify: 1) exclude Leading prepositions, qualifiers, or possessive adjectives from selecting the ADR span, to avoid inconsistency. For instance, in the sentence "`it increases my anxiety`" only `anxiety` should be annotated; and 2) annotate all relevant contexts for an ADR concept. For example, in the sentence "`I have a severe muscle pain`", "`severe muscle pain`" should be annotated (not just "`muscle pain`"). To validate the labels, two of the authors manually checked again the annotations and found some ADRs which were not detected by the annotator; also, ambiguous ADRs where identified and discussed with the medical expert. From all the annotated posts, 600 posts with 9,326 sentences contained at least one ADR which were split into training and testing as shown in Table 5.3.

Compared Methods

We compare our proposed approach to established state-of-the-art ADR detection algorithms of different types:

- **QuickUMLS [196]:** an approximate dictionary matching algorithm which relies on UMLS concepts. We used the following setting, mentioned in [196] as having best performance: *Similarity threshold* = 0.9, *Semantic types* = $[SignorSymptom, DiseaseorSyndrome, Finding, Neoplastic\ Process]$
- **CRF (Baseline)**. The Conditional Random Field Phrase Detection Model [8] trained on the manually annotated training data LC.
- **CRF+VAE (Proposed):** In our proposed approach, we train a CRF model on the expanded training data created using the Variational Auto-Encoder approach as discussed in Section 5.3.
- **BLSTM-RNN[35]:** A state-of-the-art Bidirectional Long Short Term Memory (BLSTM) recurrent neural network (RNN) trained on the manually annotated training data LC.
- **BLSTM-RNN+VAE (proposed):** We combined our proposed technique with the BLSTM-RNN phrase detection technique. This is to highlight that our method can be combined with any supervised phrase detection technique.

[5] `https://www.reddit.com/r/AskDocs/`

[6] `https://www.reddit.com/r/DiagnoseMe/`

[7] `https://www.reddit.com/r/bipolar/`

[8] `https://github.com/dat/stanford-ner`

To demonstrate the effectiveness of different strategies for augmenting the training data for ADR phrase detection, we compare our proposed approach with the following techniques:

- **CRF+SelfTraining [207]:** A simple semi-supervised learning technique, where we train a similar conditional-random field phrase detection model as described before, but we apply the trained model on a set of randomly selected unlabeled sentences from *UGC* (i.e. we used 500 samples). The sentences containing newly annotated ADRs are added to the initial training data and are used to re-train the phrase detection model.
- **CRF+Doc2vec:** CRF model trained on data expanded using an embedding-based strategy. Instead of generating new content *SC* using VAE, we use Doc2vec [111] which is inspired by word2vec [153] to find sentences similar to the labeled content *LC*.

Training

For training the Variational Autoencoder described in Section 5.3, we set the word dropout to 0.5, the learning rate to 0.001 and we used GRU [33] for both the encoder and the decoder. For labeling the newly generated sentences, we used word embeddings as described in [153]. For Twitter we used pre-trained word embeddings trained on Twitter as described in [70]. Since these pre-trained word embeddings did not perform well on the Reddit dataset, we trained a custom word embedding on all our Reddit data. We trained the skip-gram *word2vec* (300 dimension) model on the whole Reddit unlabeled collection.

5.5 Results and Discussions

Comparison with ADR Detectors

In the first experiment, we compare our approach (i.e. trained with 100% of the labeled training data with 1 sample generated for each sample in the *LC*) against different ADR detector techniques described in Section 5.4. Tables 5.4 and 5.5 report precision, and recall and F1-measure, of all the baselines in comparison to proposed approach CRF+VAE in *Twitter* and *Reddit* datasets. We make the following observations: *QuickUMLS* is outperformed by all the other methods. The result shows that dictionary based approaches are not able to cover concepts that do not have a reference in UMLS dictionary, and produce false positives by labeling irrelevant words such as “maybe”, “energy”, “condition”, “illness”, or “worse” as positive.

The difference in performance between CRF and CRF+VAE shows the advantage brought by the sentence generation (VAE) and sentence annotation step of our approach. To highlight that our method can be combined with any supervised phrase detection technique, we combined our proposed technique with the BLSTM-RNN. BLSTM-RNN outperforms CRF in Twitter dataset; note that the model was designed to detect ADRs from the Twitter dataset. The results show that independent of the methodology used

for training an ADR detector (e.g. CRF or BLSTM-RNN), expanding training data with VAE improves the overall performance. However due to the large amount of time required for training the BLSTM-RNN and the unstable prediction performance of its model on the test set [35], the remaining experiments just focus on CRF for training ADR detector.

Table 5.4: Performance of the different ADR detection techniques on the Twitter test set.

Technique	**Precision**	**Recall**	**Fscore**
QuickUMLS	.47	.34	.39
CRF	.67	.42	.51
BLSTM-RNN	.61	**.87**	.72
CRF+VAE	.68	.49	.57
BLSTM-RNN+VAE	**.71**	.85	**.77**

Table 5.5: Performance of the different ADR detection techniques on the Reddit test set.

Technique	**Precision**	**Recall**	**Fscore**
QuickUMLS	.14	.21	17
CRF	**.72**	.47	.57
BLSTM-RNN	.67	.28	.39
CRF+VAE	.69	**.52**	**.60**
BLSTM-RNN+VAE	.63	.29	.40

Effects of Training Data Size on CRF+VAE

For a given dataset (*Twitter* and *Reddit*), we created smaller subsets of the training data (i.e. 10%, 25%, 50%, 75%, 100%) to simulate the effect of limited training data availability. The subsets are randomly selected, and experiments are repeated 10 times for each size setting. We then train a CRF algorithm and different variants of our CRF+VAE (i.e. with different subsets of training data and different size of newly generated content for each labeled sample) and compare their performance. In particular, the core advantage of our approach is that we are able to generate any number of additional training data samples. Therefore, we test different settings where we generate an extra 1, 5, or 10 artificial sentences for each labeled sentence in the training set. The experiments are conducted 10 times for each setting.
Figure 5.2 summarizes the average performance achieved for *Twitter* and *Reddit* datasets. The results show that by using the VAE to expand the training data, it is possible to obtain higher F-scores for both datasets. In Addition, we can show that by increasing

the number of artificially generated samples (i.e. 5 and 10 samples), we can achieve a considerable F-score boost up to (+.17) and (+.12) for Twitter and Reddit (i.e. with just 10% of the labeled samples). We did not observe any significant improvement with more than 10 samples. This limitation is likely due to our constraint to generate sentences similar to the existing annotated sentences instead of radically new ones - a limitation chosen to allow us to perform reliable label propagation which would be hard for sentences too different. The results also show that by generating 1 sample using VAE but only using 50% of the training data, we can obtain comparable results to using the 100% of the labeled training data without VAE. When generating more training samples (i.e. 10 samples), our approach can achieve comparable performance with only the 25% of the initial labeled set. As shown in Figure 5.2, the effect of VAE expansion is greater the smaller the training data set is, thus VAE is used most efficiently to reduce the training costs of ADR detection significantly while maintaining quality. Note that all the improvements of CRF+VAE over CRF are statistically significant using paired t-test (i.e. p-value<0.05). When artificially expanding training data, recall is often improved at the cost of precision. This is demonstrated by the performance of CRF+Doc2Vec (Table 5.6). However, even using CRF+VAE (1 sample) shows higher F-score than CRF without notable loss of precision. This positive behaviour can be attributed to the larger number of positive and negative examples which helps to maintain the generalisation capabilities of the ADR detector while refining the quality of its recognition.

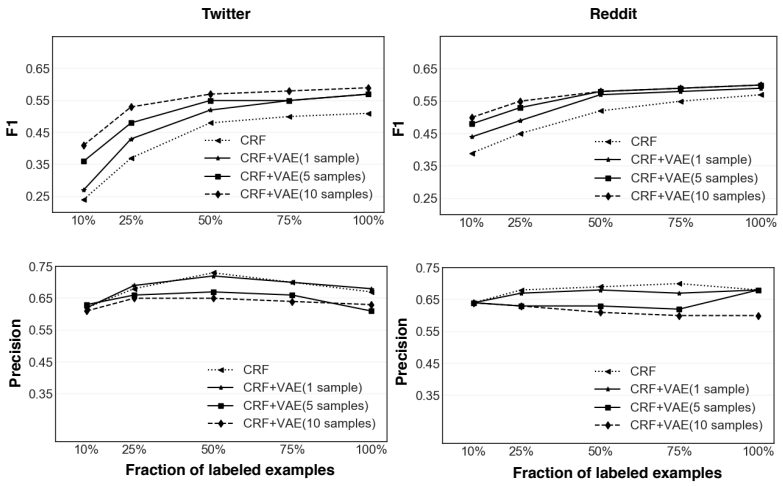


Figure 5.2: Average *F1* and *Precision* for *CRF* and *CRF+VAE* techniques, trained using different fractions of manually annotated examples and varying number of samples generated using VAE. Tested on the *Twitter* test set (on the left) and on the *Reddit* test set (on the right).

Table 5.6: Average Precision/Recall/F1 with standard deviation in parenthesis for CRF, CRF+SelfTraining, CRF+Doc2Vec and CRF+VAE on Twitter and Reddit datasets. The experiments are conducted 10 times for each setting.

Datasets	%Labeled samples	CRF	CRF+SelfTraining	CRF+Doc2Vec	CRF+VAE
Twitter	10	62(.1)/.15(.05)/.24(.07)	.65(.05)/.27(.05)/.38(.06)	.57(.09)/.30(.04)/.39(.04)	.61(.10)/.32(.04)/.41(.04)
	25	68(.06)/.25(.03)/.37(.04)	.66(.05)/.32(.02)/.43(.02)	.62(.03)/.42(.03)/.50(.03)	.65(.04)/.44(.03)/.53(.02)
	50	73(.02)/.35(.02)/.48(.02)	70(.03)/.37(.03)/.48(.03)	.65(.04)/.50(.01)/.56(.01)	.65(.01)/.51(.02)/.57(.01)
	75	70(.02)/.39(.01)/.50(.02)	68(.02)/.40(.02)/.51(.02)	.66(.02)/.52(.02)/.58(.01)	.67(.02)/.51(.03)/.58(.02)
	100	.67/.42/.51	.67/.41/.51	.61/.57/.59	.64/.56/.60
Reddit	10	.64(.06)/.28(.05)/.38(.05)	.64(.05)/.29(.05)/.40(.04)	.62(0.04)/.42(.04)/.50(0.3)	.64(.03)/.41(.04)/.50(.03)
	25	.68(.03)/.34(.03)/.45(.03)	.68(.03)/.34(.04)/.45(.03)	.61(.02)/.51(.02)/.55(.02)	.63(.02)/.48(.01)/.55(.01)
	50	.69(.02)/.42(.03)/.52(.02)	.69(.02)/.43(.04)/.53(.02)	.57(.02)/.60(.01)/.59(.01)	.61(.01)/.56(.02)/.59(.01)
	75	.70(.01)/.46(.02)/.55(.01)	.70(.01)/.46(.02)/.55(.01)	.56(.01)/.62(.01)/.59(.01)	.60(.01)/.59(.01)/.60(.01)
	100	.72/.47/.57	.71/.46/.57	.57/.64/.61	.60/.62/.61

Comparison of Data Expansion Techniques

In the third experiment, we compare the performance of CRF+VAE against the two other automatic training data expansion techniques CRF+SelfTraining and CRF+Doc2Vec. As in the previous experiment, we use 10%, 25%, 50%, 75% and 100% of the training data. For the sake of brevity, we only report the best performance[9] achieved by these techniques in Table 5.6.

CRF+SelfTraining keeps the precision high but compared to CRF+Doc2Vec and CRF+VAE, it is not able to increase the recall significantly. Its low recall can be attributed to treating some terms incorrectly as negative instance examples. This is due to relying only on the output of the trained model for labeling the training data for the next iteration. We observe that CRF+VAE achieves better precision and comparable recall to CRF+Doc2Vec with the Twitter dataset, while achieving similar performance in the Reddit dataset in terms of F-score, but with higher precision. This underlines that artificially generating new similar training sentences can outperform discovering existing similar training sentences using Doc2Vec similarity. The results show that our approach in general performs better in the Twitter dataset. This can likely be attributed to the differences in the structure between the two datasets. Each tweet contains on average 8 words, while each Reddit sentence contains on average 17 words. Also, VAEs have shown to perform better on shorter sentences [188].

5.6 Qualitative Analysis

In this section we tested CRF+VAE approach on *Twitter* and *Reddit* test sets and manually inspect all the posts containing false positive and negatives to understand the reasons for the prediction errors.

False Positives. Manual inspection of the posts reveal that most of the false positives are due to 1) Mis-recognizing *indications* as an ADR, i.e. an illness for which the drug

[9]The Self-training configuration has been run for ten iterations; we report the iteration with best performance.

has been prescribed is recognized as an adverse drug reaction [34]. For instance in the two posts *"I started effexor after having pretty severe postpartum depression"* and *"depression hurts cymbalta can help"*, `depression` is labeled as ADR even though it is an *indication*. However, `depression` commonly occur as ADR as well in other posts, which might be the cause for this error [34]; 2) Ignoring negative verbs. As an example the word `manic` in *"The only one that didn't make me manic, Wellbrutin"* and `vomiting` in *"@uclaibd I never had bleeding or vomiting just a lot of fatigue"* are detected as ADRs due to the structure of the posts. However the model was not able to distinguish the negative verbs; 3) Mislabeling ADR-related words as an ADR: For instance in the post *"temperature would start to rise, depression weakens"* the word `depression` was recognized as ADR; 4) Mistakes in manual annotation in the test data. For instance in the Tweet *"Ive had no appetite since I started on prozac"*, the annotators did not annotate *no appetite* as an ADR. However, our model was able to predict it correctly as an ADR, but due to this mistake in test data is considered a false positive.

False Negatives. False negatives are likely to occur in posts that are ambigious or overly complex. For example, in the post *"Im just wondering if its safe to take tramadol 15h after vyanse and if promethazine and melatonin would lower my chances of a seizure"* the word `seizure` was not detected as an ADR. It must be noted how, in this specific case, even human annotators debated if seizure is indeed an ADR of tramadol, or an indication of vyanse. In another example *"Am I the only one that grinds the shit out of their teeth on Vyvanse"*. The expression `grinds the shit out of their teeth` is a long description of the slang ADR *teeth grind*, which has been described in a very unstructured and informal way. This is hard to handle for phrase detectors like CRF as some level of abstraction would be necessary to deal with this.

5.7 Conclusion

In this chapter, we have demonstrated an approach for augmenting training data for detecting user-generated phrases (i.e., mentions of Adverse Drug Reactions) from social media text in a very cost-efficient manner. We introduced a technique which expands human-labeled training sets with a large number of artificially generated training samples. The benefit of our training data generation technique is greater, the smaller the manually created training data set is. Therefore, it is used most efficiently to reduce the manual training costs of ADR detection while maintaining quality (e.g., in our experiments, we can maintain quality even when reducing manually provided training data by 75%). Furthermore, we could show that our approach generally works better on Twitter data. We assume that this can be explained by Reddit forum posts using significantly richer, longer, and more complex sentences. VAEs are known to work more effectively with shorter sentences than with longer ones. This work is only one of the initial steps towards automated adverse drug effect analytics on social data. The next step would be to interpret the semantics of the extracted slang ADRs, and linking them to medical ontologies to allow for further structured analysis.

Acknowledgments

We thank Dr. Gerhard Mulder for his wonderful collaboration in annotating the Reddit dataset.

Chapter 6

Conclusion

Named Entity Recognition (NER) is an important Information Retrieval task and a key enabler for many applications (e.g., content retrieval, exploration). NER is performed using different techniques such as dictionary-based, rule-based, and machine learning-based. Supervised machine learning techniques have shown to perform the best, but only under the condition of the availability of good quality annotated training data, which is rarely and sparsely available for long-tail entities. This dissertation has focused on efficient and effective ways to improve the performance of long-tail entity recognizers. We examined different training data augmentation techniques in the context of scientific publications and user-generated content. In this chapter we summarize the main contributions made in this thesis and provide an outlook on future research directions.

6.1 Research Questions Revisited

At the beginning of this thesis, we formulated a set of research questions that were investigated in Chapters 2 to 5. In this section we will discuss the results for each research question.

Research Question 1

> *To what extent can pre-trained NER recognize long-tail entities?*

In Chapter 2, we investigated the usage of pre-trained NER for recognizing long-tail entities. We hypothesized that by using the existing pre-trained NER, which is trained on a large amount of training data, we could identify the entities mentioned in the text. We used TextRazor API [1] to return the detected entities, possibly decorated with links to the DBpedia or Freebase knowledge bases. As we get all named entities of a sentence, the result list contains many entities that have a type not explicitly related to long-tail entity types. To address this issue, we first classified the sentences in a given text into pre-defined entity types using distant supervision. Next, we used existing pre-trained NER to extract the long-tail entities from the classified sentences and assigned them the type matching the sentence class. Our proposed approach, despite its simple design, features lower training costs (compared to traditional supervised learning) and acceptable performance. The results suggested that as further improvement, there is a need to train domain-specific NERs which can detect and type long-tail entities.

Research Question 2

> *How can semantic expansion techniques and filtering heuristics be leveraged to augment training data for L-tER?*

The main challenge in the training of L-tER is the lack of training data. In Chapter 3, we have pursued a low-cost iterative approach for augmenting the training data for training a Long-tail entity extraction model. The proposed technique (TSE-NER) is based on a set of expansion strategies exploiting semantic similarity and relatedness between terms to increase the size and labeling quality of the training dataset generated from the seed terms, as well as several filtering techniques to control the noise introduced by the expansion. We conducted extensive evaluations showing that we were able to tune the technique for either higher recall or higher precision scenarios with only a small set of seed names (i.e., 5 to 100). While promising, we observed that the heuristics are prone to failure. We posed that by incrementally incorporating human feedback on the relevance of extracted entities into the training cycle of such iterative TSE-NER algorithms, we can improve the overall performance.

[1] https://www.textrazor.com/named_entity_recognition

Research Question 3

How can the collaborative feedback from human annotators be leveraged to improve L-tER?

The results in Chapter 3 showed that a limiting factor in TSE-NER was the filtering heuristics, which were amenable to failure. In Chapter 4 we proposed Coner, an approach that integrates user feedback into the TSE-NER training process to improve the overall performance. Coner allows us to still maintain the advantages of the initial design of TSE-NER and train a NER algorithm at very low cost, and then exploit the daily user interaction (like in a digital library system) for continuously improving the algorithm's performance, requiring only simple and intuitive feedback actions from the users. The experiments showed that with Coner we could decrease the noise in the filtering heuristics and increase the overall performance of the NER. The applicability of the proposed approach is currently limited. To fully leverage Coner's potential, Coner needs to be integrated into an existing production system, like a large scale digital library to receive continuous feedback from the system's users.

Research Question 4

How can deep generative models be leveraged to improve the performance of L-tER?

In Chapter 5, we use deep probabilistic models to capture the underlying structure of the data, which allows generating new training samples resembling the subset of the corpus for which human annotation is available. The newly generated samples were heuristically annotated by propagating the labels of the initial seeds. Through an extensive evaluation performed on Twitter and Reddit, we showed that our approach can reduce the need for training data and improve the overall performance of the L-tER. One limitation of our approach is its constraint to generate sentences similar to the existing annotated sentences instead of radically new ones. This was clearly shown when there was no significant improvement with more than 10 samples.

6.2 Future Work

This dissertation shows the need for novel Named Entity Recognition approaches targeting long-tail entities. We contribute novel techniques for training data augmentation that are capable of improving the performance of the long-tail entity recognizer. While we consider our results promising, we identify several directions for further investigation.

Context-dependant Word Embeddings

The techniques described in Chapter 3, 4 and 5 rely on context-independent word embeddings approaches, namely word2vec and Glove. These models do not take into account the sequential context and combine all the different senses of the word into one vector.

For example, the word "apple" refers to very different things in the following sentences, "I am eating an apple" and "I bought an Apple phone", but they will have the same word embedding vector [2]. An interesting direction for further investigation comprises the use of context-dependent word embeddings such as BERT [51], ELMO [171] and XLNET [217]. Context-dependant word embeddings result in different representations of a word depending on the context it appears and have shown to perform better than context-independent word embedding in several NLP tasks such as NER, sentence classification and sentiment analysis [3]. As future work, we plan to investigate how different word embedding models affect the overall performance of the proposed training data augmentation techniques (i.e., we initiated a research project which is currently under submission [131]).

Active Learning For Long-tail Entity Recognition

The work presented in Chapter 4 shows that it is possible to enhance the performance of L-tER by incorporating the feedback of humans while at the same keeping the cost of human supervision low. We expect that further improvement can be achieved by employing techniques to select the most informative sample for annotation. Active learning methods such as Uncertainty sampling [116] and Query by Committee (QBC) [136] are popular approaches that aim to reduce the cost of supervision by exploring the unlabeled dataset and selecting new training samples for annotation. Previous research has shown that active learning techniques are not likely to select samples that belong to the rare-class [10, 122]. However, we speculate that by fusing the output of active learning with the techniques proposed in this thesis, we could identify the most informative sample to annotate and thus increase the overall performance of the L-tER.

Dealing with Cold Start Problem

All the techniques presented in this thesis require an initial set of seed annotated samples. Our analysis in Chapter 3 showed that the selection of the seed set affects the overall performance of the L-tER. How to estimate the quality of a seed set is still an open problem. We, therefore, recommend future research to focus on the cold start problem, finding ways to select the best seed sets as a starting point, which can increase the performance of our techniques. Cold start problem has been investigated in the field of recommender systems [186, 160, 14]. We can borrow techniques from the area of recommender systems that can be applied to our problem, for instance, by exploring the usage of exploration vs. exploitation trade-offs in a multi-armed bandit problem [14]. The multi-armed bandit problem is a reinforcement learning example where we have different arms (bandits), each arm has its probability distribution of success (e.g., reward=+1 for success, or reward=0 for failure). The objective is to maximize the sum of rewards earned through a sequence of actions. As future work, it is interesting to

[2] https://towardsdatascience.com/from-word-embeddings-to-pretrained-language-models-a-new-age-in-nlp-part-2-e9af9a0bdcd9

[3] https://allennlp.org/elmo

investigate how the multi armed-bandit can solve the cold start problem when dealing with training data augmentation for long-tail entities recognition.

From Entity Extraction to Knowledge Extraction

This work is only one of the initial steps towards automated long-tail-entity extraction for English text. However, in order to be able to automatically process the ever-increasing amount of the data on the Web, there is a need for effective methods that are able to interpret the semantics of the extracted entities, link them to ontologies and taxonomies (if available), and find the relation between different entities in different languages to allow for further structured analysis. As the next step in this direction, we initiated two research projects on: 1) Multilingual Open Relation Extraction (ORE) when limited resources are available [85] (in Appendix B) and 2) Normalizing Adverse Drug Reactions (ADR) reports from user-generated content to concepts in a controlled medical vocabulary [131].

Open Relation Extraction (ORE) aims to find arbitrary relation tuples between entities in unstructured texts. Even though recent research efforts yield state-of-the-art results for the ORE task by utilizing neural models, these works solely focus on the English language. Methods were proposed to tackle the ORE task for multiple languages, yet these works fail to exploit relation patterns that are consistent over languages.

The automatic mapping of Adverse Drug Reactions (ADR) reports from user-generated content to a concept in a controlled medical vocabulary provides valuable insights for monitoring public health. While state-of-the-art deep learning-based techniques achieve impressive performance for medical concepts with large amounts of training samples, they show their limit with long-tail concepts that have no/low number of training samples. This limits their effectiveness and adaptability to the high dynamicity of laymans terminology.

Our long-term vision is to establish entity extraction, entity linking, and relation extraction techniques, which can deal with different kinds of input text (i.e., structured, noisy text, long-tail) in different languages, which will lead us towards novel exploration capabilities.

Bibliography

[1] In *English for Writing Research Papers Useful Phrases*.

[2] Slimani Abdelali et al. Education data mining: Mining moocs videos using metadata based approach. In *Information Science and Technology (CiSt)*, pages 531–534. IEEE, 2016.

[3] Takeshi Abekawa and Akiko Aizawa. Sidenoter: Scholarly paper browsing system based on pdf restructuring and text annotation. In *COLING (Demos)*, pages 136–140, 2016.

[4] Eleni Afiontzi, Giannis Kazadeis, Leonidas Papachristopoulos, Michalis Sfakakis, Giannis Tsakonas, and Christos Papatheodorou. Charting the digital library evaluation domain with a semantically enhanced mining methodology. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 125–134. ACM, 2013.

[5] Syed Rizwanuddin Ahmad. Adverse drug event monitoring at the food and drug administration: your report can make a difference. *Journal of general internal medicine*, 18(1):57–60, 2003.

[6] Akiko Aizawa. Pdfnlt. `https://github.com/KMCS-NII/PDFNLT`, 2018.

[7] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.

[8] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 344–354, 2015.

[9] Eiji Aramaki, Sachiko Maskawa, and Mizuki Morita. Twitter catches the flu: detecting influenza epidemics using twitter. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1568–1576. Association for Computational Linguistics, 2011.

[10] Josh Attenberg and Foster Provost. Why label when you can search?: alternatives to active learning for applying human resources to build classification models under extreme class imbalance. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 423–432. ACM, 2010.

[11] Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner, K Bretonnel Cohen, Karin Verspoor, Judith A Blake, et al. Concept annotation in the craft corpus. *BMC bioinformatics*, 13(1):161, 2012.

[12] Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. Cloze-driven pretraining of self-attention networks. *arXiv preprint arXiv:1903.07785*, 2019.

[13] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[14] Andrea Barraza-Urbina. The exploration-exploitation trade-off in interactive recommender systems. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 431–435. ACM, 2017.

[15] Maksim Belousov, Nikola Milosevic, William Dixon, and Goran Nenadic. Extracting adverse drug reactions and their context using sequence labelling ensembles in tac2017. *TAC*, 2019.

[16] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[17] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

[18] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[19] Thomas Bosch, Richard Cyganiak, Arofan Gregory, and Joachim Wackerow. Ddi-rdf discovery vocabulary: A metadata vocabulary for documenting research and survey data. In *LDOW*, 2013.

[20] Adrien Bougouin, Florian Boudin, and Béatrice Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. In *Int. Joint Conf. on Natural Language Processing (IJCNLP)*, pages 543–551, 2013.

[21] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, 2016.

[22] Alessandro Bozzon, Piero Fraternali, Luca Galli, and Roula Karam. Modeling crowdsourcing scenarios in socially-enabled human computation applications. *Journal on Data Semantics*, 3(3):169–188, Sep 2014.

[23] Marco Brambilla, Stefano Ceri, Emanuele Della Valle, Riccardo Volonterio, and Felix Xavier Acero Salazar. Extracting emerging knowledge from social media. In *Proceedings of the 26th International Conference on World Wide Web*, pages 795–804. International World Wide Web Conferences Steering Committee, 2017.

[24] Sabine Buchholz and Antal Van Den Bosch. Integrating seed names and ngrams for a named entity list and classifier. In *LREC*, 2000.

[25] Gully APC Burns, Pradeep Dasigi, Anita de Waard, and Eduard H Hovy. Automated detection of discourse segment and experimental types from the text of cancer pathway results sections. *Database*, 2016, 2016.

[26] Maria Elisabete Catarino and Ana Alice Baptista. Relating folksonomies with dublin core. In *Dublin Core Conference*, pages 14–22, 2008.

[27] Lauren E Charles-Smith, Tera L Reynolds, Mark A Cameron, Mike Conway, Eric HY Lau, Jennifer M Olsen, Julie A Pavlin, Mika Shigematsu, Laura C Streichert, Katie J Suda, et al. Using social media for actionable disease surveillance and outbreak management: A systematic literature review. *PloS one*, 10(10):e0139701, 2015.

[28] Brant W Chee, Richard Berlin, and Bruce Schatz. Predicting adverse drug events from personal health messages. In *AMIA Annual Symposium Proceedings*, volume 2011, page 217. American Medical Informatics Association, 2011.

[29] Chaomei Chen. CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for information Science and Technology*, 57(3):359–377, 2006.

[30] Luying Chen, Stefano Ortona, Giorgio Orsi, and Michael Benedikt. Aggregating semantic annotators. *Proceedings of the VLDB Endowment*, 6(13):1486–1497, 2013.

[31] Xilun Chen and Claire Cardie. Unsupervised multilingual word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 261–270. Association for Computational Linguistics, 2018.

[32] Yufeng Chen, Chengqing Zong, and Keh-Yih Su. A joint model to identify and align bilingual named entities. *Computational linguistics*, 39(2):229–266, 2013.

[33] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *In Proceedings of the Empirical Methods in Natural Language Processing*, 2014.

[34] Shaika Chowdhury, Chenwei Zhang, and Philip S Yu. Multi-task pharmacovigilance mining from social media posts. In *Proceedings of the 27th International Conference on World Wide Web*, pages 117–126. International World Wide Web Conferences Steering Committee, 2018.

[35] Anne Cocos, Alexander G Fiks, and Aaron J Masino. Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in twitter posts. *Journal of the American Medical Informatics Association*, 24(4):813–821, 2017.

[36] Learning Technology Standards Committee et al. Ieee standard for learning object metadata. *IEEE Standard*, 1484(1):2007–04, 2002.

[37] Companion. Companion page. In `https://sites.google.com/view/emnlp-ijcnlp2019`, 2019.

[38] IMS Global Learning Consortium. Learning resource meta-data specification. `https://www.imsglobal.org/metadata/index.html`, 2002. Accessed: 2018-02-26.

[39] Alexandru Constantin, Silvio Peroni, Steve Pettifer, David Shotton, and Fabio Vitali. The document components ontology (doco). *Semantic Web*, 7(2):167–181, 2016.

[40] Dan Cosley and Steve Lawrence. REFEREE: An open framework for practical testing of recommender systems using ResearchIndex. *Proceedings of the 28th VLDB Conference*, pages 35–46, 2002.

[41] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[42] Lei Cui, Furu Wei, and Ming Zhou. Neural open information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 407–413. Association for Computational Linguistics, 2018.

[43] Aron Culotta, Andrew McCallum, and Jonathan Betz. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 296–303. Association for Computational Linguistics, 2006.

[44] Xiang Dai, Sarvnaz Karimi, and Cecile Paris. Medication and adverse event extraction from noisy text. In *Proceedings of the Australasian Language Technology Association Workshop 2017*, pages 79–87, 2017.

[45] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys (CSUR)*, 51(1):7, 2018.

[46] Arjun Magge Ashlynn Daughton Karen O'Connor Michael Paul Graciela Gonzalez-Hernandez. Davy Weissenbacher, Abeed Sarker. Overview of the fourth social media mining for health (smm4h) shared task at acl 2019. in proceedings of the 2019 acl workshop smm4h: The 4th social media mining for health applications workshop shared task. 2019.

[47] Munmun De Choudhury, Shagun Jhaver, Benjamin Sugar, and Ingmar Weber. Social media participation in an activist movement for racial equality. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[48] Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366. ACM, 2013.

[49] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, 2017.

[50] Jan Deriu, Aurelien Lucchi, Valeria De Luca, Aliaksei Severyn, Simon Müller, Mark Cieliebak, Thomas Hofmann, and Martin Jaggi. Leveraging large amounts of weakly supervised data for multi-language sentiment classification. In *Proceedings of the 26th international conference on world wide web*, pages 1045–1052. International World Wide Web Conferences Steering Committee, 2017.

[51] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[52] Fabiano A Dorça, Vitor C Carvalho, Miller M Mendes, Rafael D Araújo, Hiran N Ferreira, and Renan G Cattelan. An approach for automatic and dynamic analysis of learning objects repositories through ontologies and data mining techniques for supporting personalized recommendation of content in adaptive and intelligent educational systems. In *Advanced Learning Technologies (ICALT)*, pages 514–516. IEEE, 2017.

[53] Heidrun Dorgeloh and Anja Wanner. Formulaic argumentation in scientific discourse. *Formulaic Language: Volume 2. Acquisition, loss, psychological reality, and functional explanations*, 83:523, 2009.

[54] Eric Duval, Erwin Vervaet, Bart Verhoeven, Koen Hendrikx, Kris Cardinaels, Hendrik Olivié, Eddy Forte, Florence Haenni, Ken Warkentyne, M Wentland Forte, et al. Managing digital educational resources with the ariadne metadata system. *Journal of Internet cataloging*, 3(2-3):145–171, 2000.

[55] Tome Eftimov, Barbara Koroušić Seljak, and Peter Korošec. A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations. *PloS one*, 12(6):e0179488, 2017.

[56] Samhaa R El-Beltagy and Ahmed Rafea. Kp-miner: A keyphrase extraction system for english and arabic documents. *Information Systems*, 34(1):132–144, 2009.

[57] Kinda El Maarry, Ulrich Güntzer, and Wolf-Tilo Balke. A majority of wrongs doesn't make it right-on crowdsourcing quality for skewed domain tasks. In *International Conference on Web Information Systems Engineering*, pages 293–308. Springer, 2015.

[58] José Esquivel, Dyaa Albakour, Miguel Martinez, David Corney, and Samir Moussa. On the long-tail entities in news. In *European Conference on Information Retrieval*, pages 691–697. Springer, 2017.

[59] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

[60] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[61] Manaal Faruqui and Shankar Kumar. Multilingual open relation extraction using cross-lingual projection. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1351–1356. Association for Computational Linguistics, 2015.

[62] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. 1997.

[63] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. 1997.

[64] Katrin Fundel, Robert Küffner, and Ralf Zimmer. Relexrelation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2006.

[65] Christopher Funk, William Baumgartner, Benjamin Garcia, Christophe Roeder, Michael Bada, K Bretonnel Cohen, Lawrence E Hunter, and Karin Verspoor.

Large-scale biomedical concept recognition: an evaluation of current automatic annotators and their parameters. *BMC bioinformatics*, 15(1):59, 2014.

[66] Pablo Gamallo and Marcos Garcia. Multilingual open information extraction. In *Portuguese Conference on Artificial Intelligence*, pages 711–722. Springer, 2015.

[67] Yolanda Gil, Varun Ratnakar, and Daniel Garijo. Ontosoft: Capturing scientific software metadata. In *Proceedings of the 8th International Conference on Knowledge Capture*, page 32. ACM, 2015.

[68] H Glaser and I Millard. Knowledge-enabled research support: Rkbexplorer. com. *Proceedings of Web Science*, 2009.

[69] Balazs Godeny. Rule based product name recognition and disambiguation. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 858–860. IEEE, 2012.

[70] Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153, 2015.

[71] Sean Goldberg, Daisy Zhe Wang, and Christan Grant. A probabilistically integrated system for crowd-assisted text labeling and extraction. *Journal of Data and Information Quality (JDIQ)*, 8(2):10, 2017.

[72] Sean Louis Goldberg, Daisy Zhe Wang, and Tim Kraska. Castle: crowd-assisted system for text labeling and extraction. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[73] Archana Goyal, Vishal Gupta, and Manish Kumar. Recent named entity recognition and classification techniques: a systematic review. *Computer Science Review*, 29:21–43, 2018.

[74] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[75] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[76] Tudor Groza. Using typed dependencies to study and recognise conceptualisation zones in biomedical literature. *PloS one*, 8(11):e79570, 2013.

[77] Thomas R Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5):907–928, 1995.

[78] Philip J. Guo, Juho Kim, and Rob Rubin. How video production affects student engagement: An empirical study of mooc videos. In *ACM Conf. on Learning @ Scale Conference*, L@S '14, pages 41–50, New York, NY, USA, 2014. ACM.

[79] Sonal Gupta and Christopher D Manning. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *In Proceedings of IJCNLP*. Citeseer, 2011.

[80] Ivan Habernal and Miloslav KonopíK. Swsnl: semantic web search using natural language. *Expert Systems with Applications*, 40(9):3649–3664, 2013.

[81] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14):i37–i48, 2017.

[82] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

[83] Rave Harpaz, William DuMouchel, Nigam H Shah, David Madigan, Patrick Ryan, and Carol Friedman. Novel data-mining methodologies for adverse drug event discovery and analysis. *Clinical Pharmacology & Therapeutics*, 91(6):1010–1021, 2012.

[84] Z. Harris. Distributional Structure. *Word*, 10:146–162, 1954.

[85] Tom Harting, Sepideh Mesbah, and Christoph Lofi. Language-consistent open relation extraction: from multilingual text corpora. In *The Web Conference (To appear)*, 2020.

[86] Zhe He, Zhiwei Chen, Sanghee Oh, Jinghui Hou, and Jiang Bian. Enriching consumer health vocabulary through mining a social q&a site: A similarity-based approach. *Journal of biomedical informatics*, 69:75–85, 2017.

[87] Afrida Helen, Ayu Purwarianti, and Dwi H Widyantoro. Rhetorical sentences classification based on section class and title of paper for experimental technical papers. *Journal of ICT Research and Applications*, 9(3):288–310, 2015.

[88] Apirak Hoonlor, Boleslaw K Szymanski, and Mohammed J Zaki. Trends in computer science research. *Communications of the ACM*, 56(10):74–83, 2013.

[89] Tianran Hu, Haoyuan Xiao, Jiebo Luo, and Thuy-vy Thi Nguyen. What the language you tweet says about your occupation. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[90] Yuheng Hu, Shelly Farnham, and Kartik Talamadupula. Predicting user engagement on twitter with real-world events. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM). AAAI*, 2015.

[91] Simon Hudson, Li Huang, Martin S Roth, and Thomas J Madden. The influence of social media interactions on consumer–brand relationships: A three-country study of brand perceptions and marketing behaviors. *International Journal of Research in Marketing*, 33(1):27–41, 2016.

[92] Trung Huynh, Yulan He, Alistair Willis, and Stefan Rueger. Adverse drug reaction classification with deep neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 877–887, 2016.

[93] F Ilievski. Identity of long-tail entities in text. 2019.

[94] Petra Isenberg, Tobias Isenberg, Michael Sedlmair, Jian Chen, and Torsten Möller. Visualization as Seen Through its Research Paper Keywords. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):771–780, 2017.

[95] Shengbin Jia, Yang Xiang, and Xiaojun Chen. Supervised neural models revitalize the open relation extraction. *CoRR*, abs/1809.09408, 2018.

[96] Shan Jiang, Ana Alves, Filipe Rodrigues, Joseph Ferreira, and Francisco C Pereira. Mining point-of-interest data from social networks for urban land use classification and disaggregation. *Computers, Environment and Urban Systems*, 53:36–46, 2015.

[97] Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, 2018.

[98] Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. Cadec: A corpus of adverse drug event annotations. *Journal of biomedical informatics*, 55:73–81, 2015.

[99] Payam Karisani and Eugene Agichtein. Did you really just have a heart attack?: Towards robust detection of personal health mentions in social media. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 137–146. International World Wide Web Conferences Steering Committee, 2018.

[100] Tomi Kauppinen, Alkyoni Baglatzi, and Carsten Keßler. Linked science: interconnecting scientific assets. *Data Intensive Science. CRC Press, USA (forthcoming 2012)*, 2012.

[101] Mayank Kejriwal and Pedro Szekely. Information extraction in illicit web domains. In *Proceedings of the 26th International Conference on World Wide Web*, pages 997–1006. International World Wide Web Conferences Steering Committee, 2017.

[102] Daniel Kershaw, Matthew Rowe, and Patrick Stacey. Towards modelling language innovation acceptance in online social networks. In *Proceedings of the Ninth ACM*

International Conference on Web Search and Data Mining, pages 553–562. ACM, 2016.

[103] Hanan Khalil and Martin Ebner. Moocs completion rates and possible methods to improve retention-a literature review. In *EdMedia: World Conference on Educational Media and Technology*, pages 1305–1313. Association for the Advancement of Computing in Education (AACE), 2014.

[104] Masayu Leylia Khodra, Dwi H Widyantoro, EA Aziz, and Riyanto T Bambang. Information extraction from scientific paper using rhetorical classifier. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, pages 1–5. IEEE, 2011.

[105] Juho Kim, Philip J Guo, Daniel T Seaton, Piotr Mitros, Krzysztof Z Gajos, and Robert C Miller. Understanding in-video dropouts and interaction peaks inonline lecture videos. In *Conf. on Learning@ scale conference*, pages 31–40. ACM, 2014.

[106] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *stat*, 1050:1, 2014.

[107] René F Kizilcec, Kathryn Papadopoulos, and Lalida Sritanyaratana. Showing face in video instruction: effects on information retention, visual attention, and affect. In *SIGCHI Conf. on human factors in computing systems*, pages 2095–2102. ACM, 2014.

[108] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*, 2018.

[109] Jonathan Koren, Yi Zhang, and Xue Liu. Personalized interactive faceted search. *Proceeding of the 17th international conference on World Wide Web - WWW '08*, pages 477–485, 2008.

[110] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Int. Conf. on Machine Learning*, volume 951, pages 282–289, 2001.

[111] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.

[112] Géraud Le Falher, Aristides Gionis, and Michael Mathioudakis. Where is the soho of rome? measures and algorithms for finding similar neighborhoods in cities. In *AAAI Conference on Web and Social Media*, 2015.

[113] Robert Leaman, Laura Wojtulewicz, Ryan Sullivan, Annie Skariah, Jian Yang, and Graciela Gonzalez. Towards internet-age pharmacovigilance: extracting adverse drug reactions from user posts to health-related social networks. In *Proceedings of the 2010 workshop on biomedical natural language processing*, pages 117–125. Association for Computational Linguistics, 2010.

[114] Kathy Lee, Ashequl Qadir, Sadid A Hasan, Vivek Datla, Aaditya Prakash, Joey Liu, and Oladimeji Farri. Adverse drug event detection in tweets with semi-supervised convolutional neural networks. In *Proceedings of the 26th International Conference on World Wide Web*, pages 705–714. International World Wide Web Conferences Steering Committee, 2017.

[115] David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pages 148–156. Elsevier, 1994.

[116] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR94*, pages 3–12. Springer, 1994.

[117] Lishuang Li, Wenting Fan, and Degen Huang. A two-phase bio-ner system based on integrated classifiers and multiagent strategy. *IEEE/ACM transactions on computational biology and bioinformatics*, 10(4):897–904, 2013.

[118] Nan Li, Lukasz Kidzinski, Patrick Jermann, and Pierre Dillenbourg. How do in-video interactions reflect perceived video difficulty? In *European MOOCs Stakeholder Summit*, number EPFL-CONF-207968, pages 112–121. PAU Education, 2015.

[119] Nan Li, Łukasz Kidziński, Patrick Jermann, and Pierre Dillenbourg. Mooc video interaction patterns: What do they tell us? In *Design for teaching and learning in a networked world*, pages 197–210. Springer, 2015.

[120] Maria Liakata, Simone Teufel, Advaith Siddharthan, and Colin Batchelor. Corpora for the conceptualisation and zoning of scientific papers. 2010.

[121] Nut Limsopatham and Nigel Collier. Normalising medical concepts in social media texts by learning semantic representation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1014–1023, 2016.

[122] Christopher H Lin, Mausam Mausam, and Daniel S Weld. Active learning with unbalanced classes and example-generation queries. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*, 2018.

[123] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Neural relation extraction with multi-lingual attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 34–43, 2017.

[124] Mario Lipinski, Kevin Yao, Corinna Breitinger, Joeran Beel, and Bela Gipp. Evaluation of header metadata extraction approaches and tools for scientific pdf documents. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 385–386. ACM, 2013.

[125] Avishay Livne, Matthew P Simmons, Eytan Adar, and Lada a Adamic. The Party is Over Here : Structure and Content in the 2010 Election. *October*, 161(3):201–208, 2010.

[126] Christoph Lofi. Measuring semantic similarity and relatedness with distributional and knowledge-based approaches. *Information and Media Technologies*, 10(3):493–501, 2015.

[127] Christoph Lofi. Measuring Semantic Similarity and Relatedness with Distributional and Knowledge-based Approaches. *Database Society of Japan (DBSJ) Journal*, 14(3):1–9, 2016.

[128] F Loizides and B Schmidt. Identifying and improving dataset references in social sciences full texts. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, page 105, 2016.

[129] P. Lopez. GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications. In *European Conference on Digital Library (ECDL)*, Corfu, Greece, 2009.

[130] Patrice Lopez. Grobid: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *International conference on theory and practice of digital libraries*, pages 473–474. Springer, 2009.

[131] Emmanouil Manousogiannis, Sepideh Mesbah, Selene Baez, Alessandro Bozzon, and Robert Jan Sips. A shot in the dark: Normalizing long-tail adverse drug reaction mentions in social media. In *Journal of the American Medical Informatics Association (JAMIA) Journal (under review)*, 2020.

[132] M Marrero and J Urbano. A semi-automatic and low-cost method to learn patterns for named entity recognition. *Natural Language Engineering*, pages 1–37, 2017.

[133] George Mathew, Amritanshu Agarwal, and Tim Menzies. Trends in topics at SE conferences (1993-2013). *arXiv preprint arXiv:1608.08100*, 2016.

[134] Mausam. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 4074–4077. AAAI Press, 2016.

[135] Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 523–534, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[136] Andrew Kachites McCallumzy and Kamal Nigamy. Employing em and pool-based active learning for text classification. In *Proc. International Conference on Machine Learning (ICML)*, pages 359–367. Citeseer, 1998.

[137] Sepideh Mesbah, Alessandro Bozzon, Christoph Lofi, and Geert-Jan Houben. Describing data processing pipelines in scientific publications for big data injection. In *Proceedings of the 1st Workshop on Scholarly Web Mining*, pages 1–8. ACM, 2017.

[138] Sepideh Mesbah, Alessandro Bozzon, Christoph Lofi, and Geert-Jan Houben. Describing data processing pipelines in scientific publications for big data injection. In *Workshop on Scholary Web Mining (SWM)*, Cambridge, UK, feb 2017.

[139] Sepideh Mesbah, Alessandro Bozzon, Christoph Lofi, and Geert-Jan Houben. Describing data processing pipelines in scientific publications for big data injection. In *Proceedings of the 1st Workshop on Scholarly Web Mining*, pages 1–8. ACM, 2017.

[140] Sepideh Mesbah, Alessandro Bozzon, Christoph Lofi, and Geert-Jan Houben. Describing Data Processing Pipelines in Scientific Publications for Big Data Injection. In *WSDM Workshop on Scholary Web Mining (SWM)*, Cambridge, UK, 2017.

[141] Sepideh Mesbah, Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Facet embeddings for explorative analytics in digital libraries. In *Int. Conf. on Theory and Practice of Digital Libraries (TPDL)*, Thessaloniki, Greece, sep 2017.

[142] Sepideh Mesbah, Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Facet embeddings for explorative analytics in digital libraries. In *International Conference on Theory and Practice of Digital Libraries*, pages 86–99. Springer, 2017.

[143] Sepideh Mesbah, Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Semantic annotation of data processing pipelines in scientific publications. In *European Semantic Web Conference*, pages 321–336. Springer, 2017.

[144] Sepideh Mesbah, Kyriakos Fragkeskos, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Semantic Annotation of Data Processing Pipelines in Scientific Publications. In *Extended Semantic Web Conference(ESWC)*, 2017.

[145] Sepideh Mesbah, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Smartpub: A platform for long-tail entity extraction from scientific publications. In *The Web Conf.*, 2018.

[146] Sepideh Mesbah, Christoph Lofi, Manuel Valle Torre, Alessandro Bozzon, and Geert-Jan Houben. Tse-ner: An iterative approach for long-tail entity extraction in scientific publications. In *International Semantic Web Conference*, pages 127–143. Springer, 2018.

[147] Sepideh Mesbah, Christoph Lofi, M.V. Torre, Alessandro Bozzon, and Geert-Jan Houben. Tse-ner: An iterative approach for long-tail entity extraction in scientific publications. In *17th International Semantic Web Conference*, pages 127–143. Springer, 2018.

[148] Sepideh Mesbah, Jie Yang, Robert-Jan Sips, Manuel Valle Torre, Christoph Lofi, Alessandro Bozzon, and Geert-Jan Houben. Training data augmentation for detecting adverse drug reactions in user-generated content. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2349–2359, 2019.

[149] Yishu Miao, Lei Yu, and Phil Blunsom. Neural variational inference for text processing. In *International Conference on Machine Learning*, pages 1727–1736, 2016.

[150] Zulfat Miftahutdinov and Elena Tutubalina. End-to-end deep framework for disease named entity recognition using social media data. In *2017 IEEE 30th Neumann Colloquium (NC)*, pages 000047–000052. IEEE, 2017.

[151] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Conf. on empirical methods in natural language processing*, 2004.

[152] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR '13*, 2013.

[153] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[154] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.

[155] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.

[156] Sergio Miranda and Pierluigi Ritrovato. Automatic extraction of metadata from learning objects. In *Intelligent Networking and Collaborative Systems (INCoS)*, pages 704–709. IEEE, 2014.

[157] Tanushree Mitra, Scott Counts, and James W Pennebaker. Understanding anti-vaccination attitudes in social media. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[158] Knud Möller, Tom Heath, Siegfried Handschuh, and John Domingue. Recipes for semantic web dog foodthe eswc and iswc metadata projects. In *The Semantic Web*, pages 802–815. Springer, 2007.

[159] Syed Agha Muhammad and Kristof Van Laerhoven. Duke: A solution for discovering neighborhood patterns in ego networks. In *The 9th International AAAI Conference on Web and Social Media (ICWSM), Oxford, England*, volume 5, page 2015, 2015.

[160] Mona Nasery, Matthias Braunhofer, and Francesco Ricci. Recommendations with optimal combination of feature-based and item-based preferences. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*, pages 269–273. ACM, 2016.

[161] Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48. Association for Computational Linguistics, 2015.

[162] Azadeh Nikfarjam, Abeed Sarker, Karen Oconnor, Rachel Ginn, and Graciela Gonzalez. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, 22(3):671–681, 2015.

[163] Chikashi Nobata, Satoshi Sekine, Hitoshi Isahara, and Ralph Grishman. Summarization system integrated with named entity tagging and ie pattern discovery.

[164] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *ICWSM*, 2014.

[165] Francesco Osborne, Hélène de Ribaupierre, and Enrico Motta. Techminer: extracting technologies from academic publications. In *European Knowledge Acquisition Workshop*, pages 463–479. Springer, 2016.

[166] Francesco Osborne, Hélène de Ribaupierre, and Enrico Motta. Techminer: Extracting technologies from academic publications. 2016.

[167] Liangming Pan, Xiaochen Wang, Chengjiang Li, Juanzi Li, and Jie Tang. Course concept extraction in moocs via embedding-based graph propagation. In *Int. Joint Conference on Natural Language Processing*, volume 1, pages 875–884, 2017.

[168] Aditya Parameswaran, Hector Garcia-Molina, and Anand Rajaraman. Towards the web of concepts: Extracting concepts from large datasets. *VLDB Endowment*, 3(1-2):566–577, 2010.

[169] EU Parliament. Opening up education: Innovative teaching and learning for all through new technologies and open educational resources. *Communication from the commission to the European Parliament*, 2013.

[170] Silvio Peroni and David Shotton. Fabio and cito: ontologies for describing bibliographic resources and citations. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17:33–43, 2012.

[171] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[172] Lizhen Qu, Gabriela Ferraro, Liyuan Zhou, Weiwei Hou, and Timothy Baldwin. Named entity recognition for novel types by transfer learning. In *EMNLP*, 2016.

[173] Changqin Quan, Meng Wang, and Fuji Ren. An unsupervised text mining method for relation extraction from biomedical literature. *PloS one*, 9(7):e102039, 2014.

[174] Alexandra Pomares Quimbaya, Alejandro Sierra Múnera, Rafael Andrés González Rivera, Julián Camilo Daza Rodríguez, Oscar Mauricio Muñoz Velandia, Angel Alberto Garcia Peña, and Cyril Labbé. Named entity recognition over electronic health records through a combined dictionary-based approach. *Procedia Computer Science*, 100:55–61, 2016.

[175] Khmael Rakm Rahem and Nazlia Omar. Rule-based named entity recognition for drug-related crime news documents. *Journal of Theoretical & Applied Information Technology*, 77(2), 2015.

[176] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In *Advances in neural information processing systems*, pages 3236–3246, 2017.

[177] Ridho Reinanda, Edgar Meij, and Maarten de Rijke. Document filtering for long-tail entities. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 771–780. ACM, 2016.

[178] Álvaro Rodrigo, Joaquín Pérez-Iglesias, Anselmo Peñas, Guillermo Garrido, and Lourdes Araujo. Answering questions about european legislation. *Expert Systems with Applications*, 40(15):5811–5816, 2013.

[179] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*, pages 1–20.

[180] Almudena Ruiz-Iniesta and Oscar Corcho. A review of ontologies for describing scholarly and scientific documents. In *SePublica*, 2014.

[181] Ruhi Sarikaya. The technology behind personal digital assistants: an overview of the system architecture and key components. *IEEE Signal Processing Magazine*, 34(1):67–81, 2017.

[182] Abeed Sarker, Rachel Ginn, Azadeh Nikfarjam, Karen O'Connor, Karen Smith, Swetha Jayaraman, Tejaswi Upadhaya, and Graciela Gonzalez. Utilizing social media data for pharmacovigilance: A review. *Journal of biomedical informatics*, 54:202–212, 2015.

[183] Abeed Sarker, Rachel Ginn, Azadeh Nikfarjam, Karen OConnor, Karen Smith, Swetha Jayaraman, Tejaswi Upadhaya, and Graciela Gonzalez. Utilizing social media data for pharmacovigilance: a review. *Journal of biomedical informatics*, 54:202–212, 2015.

[184] Abeed Sarker and Graciela Gonzalez. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of biomedical informatics*, 53:196–207, 2015.

[185] Bahar Sateli and René Witte. What's in this paper?: Combining rhetorical entities with linked open data for semantic literature querying. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1023–1028. ACM, 2015.

[186] Martin Saveski and Amin Mantrach. Item cold-start recommendations: learning local collective embeddings. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 89–96. ACM, 2014.

[187] Julian Seitner, Christian Bizer, Kai Eckert, Stefano Faralli, Robert Meusel, Heiko Paulheim, and Simone Paolo Ponzetto. A large database of hypernymy relations extracted from the web. In *LREC*, 2016.

[188] Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 627–637, 2017.

[189] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[190] Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 589. Association for Computational Linguistics, 2004.

[191] Kumar Shubankar, AdityaPratap Singh, and Vikram Pudi. A frequent keyword-set based algorithm for topic modeling and clustering of research papers. In *Data Mining and Optimization (DMO), 2011 3rd Conference on*, pages 96–102. IEEE, 2011.

[192] Tarique Siddiqui, Xiang Ren, Aditya Parameswaran, and Jiawei Han. Facetgist: Collective extraction of document facets in large technical corpora. In *Int. Conf. on Information and Knowledge Management*, pages 871–880. ACM, 2016.

[193] Tarique Siddiqui, Xiang Ren, Aditya Parameswaran, and Jiawei Han. FacetGist: Collective Extraction of Document Facets in Large Technical Corpora. *Proceedings CIKM 2016*, 2016.

[194] Thiago H Silva, Pedro OS de Melo, Jussara Almeida, Mirco Musolesi, and Antonio Loureiro. You are what you eat (and drink): Identifying cultural boundaries by analyzing food & drink habits in foursquare. *arXiv preprint arXiv:1404.1009*, 2014.

[195] UmrinderPal Singh, Vishal Goyal, and Gurpreet Singh Lehal. Named entity recognition system for urdu. In *Proceedings of COLING 2012*, pages 2507–2518, 2012.

[196] Luca Soldaini and Nazli Goharian. Quickumls: a fast, unsupervised approach for medical concept extraction. In *MedIR workshop, sigir*, 2016.

[197] Min Song, Go Eun Heo, and Su Yeon Kim. Analyzing topic evolution in bioinformatics: investigation of dynamics of the field with conference data in DBLP. *Scientometrics*, 101(1):397–428, 2014.

[198] Min Song, Hwanjo Yu, and Wook-Shin Han. Developing a hybrid dictionary-based bio-entity recognition technique. *BMC medical informatics and decision making*, 15(1):S9, 2015.

[199] Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 885–895, 2018.

[200] Chen-Tse Tsai, Gourab Kundu, and Dan Roth. Concept-based analysis of scientific literature. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 1733–1738. ACM, 2013.

[201] Chen-Tse Tsai, Gourab Kundu, and Dan Roth. Concept-based analysis of scientific literature. *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13*, pages 1733–1738, 2013.

[202] Richard Tzong-Han Tsai, Shih-Hung Wu, Wen-Chi Chou, Yu-Chun Lin, Ding He, Jieh Hsiang, Ting-Yi Sung, and Wen-Lian Hsu. Various criteria in the evaluation of biomedical named entity recognition. *BMC bioinformatics*, 7(1):92, 2006.

[203] Eugene Tseytlin, Kevin Mitchell, Elizabeth Legowski, Julia Corrigan, Girish Chavan, and Rebecca S Jacobson. Noble–flexible concept recognition for large-scale biomedical natural language processing. *BMC bioinformatics*, 17(1):32, 2016.

[204] Alexandros Tsironis, Christos Katsanos, and Michail Xenos. Comparative usability evaluation of three popular mooc platforms. In *Global Engineering Education Conference (EDUCON), 2016 IEEE*, pages 608–612. IEEE, 2016.

[205] Suppawong Tuarob, Sumit Bhatia, Prasenjit Mitra, and C Lee Giles. Algorithmseer: A system for extracting and searching for algorithms in scholarly big data. *IEEE Transactions on Big Data*, 2(1):3–17, 2016.

[206] Suppawong Tuarob, Sumit Bhatia, Prasenjit Mitra, and C Lee Giles. Algorithmseer: A system for extracting and searching for algorithms in scholarly big data. *IEEE Transactions on Big Data*, 2(1):3–17, 2016.

[207] Yu Usami, Han-Cheol Cho, Naoaki Okazaki, and Jun'ichi Tsujii. Automatic acquisition of huge training data for bio-medical named entity recognition. In *Proceedings of BioNLP 2011 Workshop*, pages 65–73. Association for Computational Linguistics, 2011.

[208] Frans Van der Sluis, Jasper Ginn, and Tim Van der Zee. Explaining student behavior at scale: The influence of video complexity on student dwelling time. In *ACM Conf. on Learning @ Scale*, L@S '16, pages 51–60, New York, NY, USA, 2016. ACM.

[209] Laurent Vannini and Hervé Le Crosnier. *Net.lang: Towards the Multilingual Cyberspace*. C & F Editions, 2012.

[210] Daniel Vliegenthart, Sepideh Mesbah, Christoph Lofi, Akiko Aizawa, and Alessandro Bozzon. Coner: A collaborative approach for long-tail named entity recognition in scientific publications. In *International Conference on Theory and Practice of Digital Libraries*, pages 3–17. Springer, 2019.

[211] Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. Perspectives on crowdsourcing annotations for natural language processing. *Language resources and evaluation*, 47(1):9–31, 2013.

[212] Xiaozhi Wang, Xu Han, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Adversarial multi-lingual neural relation extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1156–1166, 2018.

[213] Yaqing Wang and Quanming Yao. Few-shot learning: A survey. *arXiv preprint arXiv:1904.05046*, 2019.

[214] Anbang Xu, Haibin Liu, Liang Gou, Rama Akkiraju, Jalal Mahmud, Vibha Sinha, Yuheng Hu, and Mu Qiao. Predicting perceived brand personality with social media. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[215] Canwen Xu, Jing Li, Xiangyang Luo, Jiaxin Pei, Chenliang Li, and Donghong Ji. Dlocrl: A deep learning pipeline for fine-grained location recognition and linking in tweets. In *The World Wide Web Conference*, pages 3391–3397. ACM, 2019.

[216] Yazhou Yang and Marco Loog. Active learning using uncertainty information. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2646–2651. IEEE, 2016.

[217] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[218] Zhijun Yin, You Chen, Daniel Fabbri, Jimeng Sun, and Bradley Malin. # prayfordad: Learning the semantics behind why social media users disclose health information. In *Tenth International AAAI Conference on Web and Social Media*, 2016.

[219] FM Zanzotto and Marco Pennacchiotti. Language evolution in social media: a preliminary study. *LINGUISTICA ZERO*, 2012.

[220] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762, 2015.

[221] Qing T Zeng and Tony Tse. Exploring and developing consumer health vocabularies. *Journal of the American Medical Informatics Association*, 13(1):24–29, 2006.

[222] Sheng Zhang, Kevin Duh, and Benjamin Van Durme. Mt/ie: Cross-lingual open information extraction with neural sequence-to-sequence models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 64–70, 2017.

[223] Sheng Zhang, Kevin Duh, and Benjamin Van Durme. Selective decoding for cross-lingual open information extraction. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 832–842, 2017.

[224] Ziqi Zhang and Fabio Ciravegna. Named entity recognition for ontology population using background knowledge from wikipedia. In *Ontology Learning and Knowledge Discovery Using the Web: Challenges and Recent Advances*, pages 79–104. IGI Global, 2011.

[225] Alisa Zhila and Alexander Gelbukh. Comparison of open information extraction for english and spanish. In *19th Annual International Conference Dialog*, pages 714–722, 2013.

[226] Alisa Zhila and Alexander Gelbukh. Open Information Extraction from real Internet texts in Spanish using constraints over part-of-speech sequences: Problems of the method, their causes, and ways for improvement. *Revista signos*, 49:119 – 142, 03 2016.

[227] Jingbo Zhu, Huizhen Wang, Benjamin K Tsou, and Matthew Ma. Active learning with sampling by uncertainty and density for data annotations. *IEEE Transactions on audio, speech, and language processing*, 18(6):1323–1331, 2010.

[228] Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K Tsou. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 1137–1144. Association for Computational Linguistics, 2008.

Appendix A

SmartPub: A Platform for Scientific Entity Exploration

This appendix introduces a novel web-based platform that supports the exploration and visualization of extracted long-tail entities.

Abstract

This demo presents SmartPub, a novel web-based platform that supports the exploration and visualization of *shallow meta-data* (e.g., author list, keywords) and *deep meta-data* – long tail *named entities* which are rare, and often relevant only in specific knowledge domain – from scientific publications. The platform collects documents from different sources (e.g. DBLP and Arxiv), and extracts the domain-specific named entities from the text of the publications using Named Entity Recognizers (NERs) which we can train with minimal human supervision even for rare entity types. The platform further enables the interaction with the Crowd for filtering purposes or training data generation, and provides extended visualization and exploration capabilities. SmartPub will be demonstrated using sample collection of scientific publications focusing on the computer science domain and will address the entity types Dataset (i.e. dataset presented or used in a publication), and Methods (i.e. algorithms used to create/enrich/analyse a data set).

A.1 Introduction

For years, online digital libraries like the ACM Digital Library, IEEE Explore, ArXiv, etc. provided search functionalities for exploring academic publications, and have thus become a fundamental part of modern research processes. However the retrieval functionality of current systems are often limited to searching on *shallow* meta-data such as the title, the authors, keywords. They are usually not designed to support the analysis of *deep* meta-data such as topics of domain-specific interests like used datasets or algorithms relevant for scientific computer science publications. While such systems exist for some domains like medicine or biology, the costs for obtaining deep meta-data are generally prohibitive for wide-spread application.

Discovering *deep* meta-data from scientific publications could enable complex entity-centric queries. For instance, a researcher in the field of machine learning could be interested in a query like: *discovering the state of the art image classification research methods that have been successfully applied to the Imagenet dataset.* For answering the query above, a system requires to have access to entities such as the dataset used (e.g Imagenet), the research methods that have been applied on the datasets (e.g. LSTM neural network), etc. The automatic recognition and typing of such named entities rely either on supervised machine learning models, trained on expensive type-labeled data produced by human annotators or the generation of labeled training data from knowledge bases which is not suitable for long-tail entity types that are not very representative in knowledge bases.

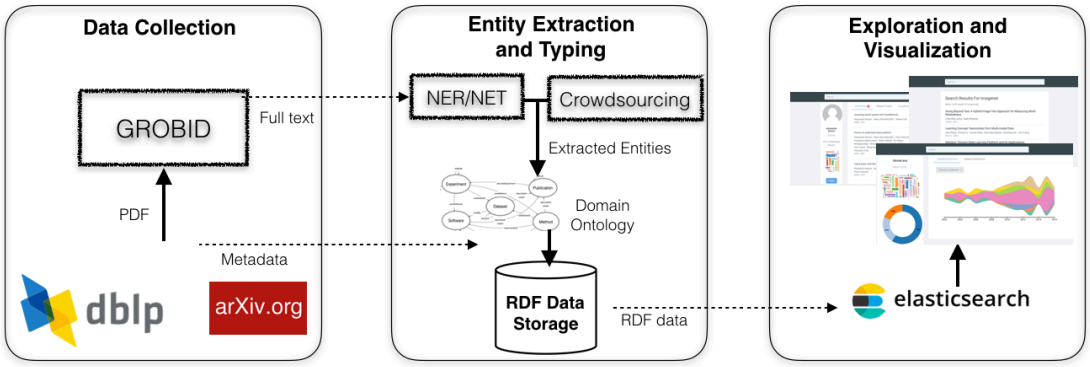


Figure A.1: Architecture of the `SmartPub` platform.

Contribution. In this demo we introduce `SmartPub`, a web-based platform that extracts long-tail entity types from scientific publication based on minimal human input, namely a small seed set of instances for the targeted entity type. Furthermore it supports the exploration and visualization of *deep* meta-data of scientific publications, i.e. meta-data able to represent domain-specific properties and aspects in which a document can be considered and understood within its (research) domain.

Users of the demo can interactively explore and visualize a collection of scientific computer science publications, by e.g. browsing for specific entities, tracking trends, discovering central concepts, or explore the usage of given entities over time. An example of the demonstration is available as a video screencast at the following address: `https://youtu.be/zLLMwOT5sZc`.

Paper Organisation. The remainder of the paper is structured as follows. Section A.2 describes the architecture of the SmartPub system, detailing its components and provided functionality. Finally, Section A.3 describes the demonstration provided to conference attendees.

A.2 The SmartPub System

The architecture of the SmartPub system is depicted in Figure E.3, where three major components are highlighted. The *Data Collection* is responsible for the retrieval of full texts and standard metadata (e.g. title, authors) of scientific publications. The *Entity Extraction* component focuses on the extraction of domain-specific entities from the publications' text, and builds a knowledge repository based on a pre-defined domain ontology. Finally, the *Exploration and Visualization* component offers user interfaces for exploration of the publications in the collection based on the extracted entities.

Data Collection

In the current implementation the data collection component retrieves scholarly data from DBLP[1] (a computer science digital library) and ArXiv[2]. For each paper, DBLP provides an XML entry that contains bibliographic meta-data (i.e. title, author names, year of publication) as well as the DOI url from which the publications' PDF can be retrieved. ArXiv offers open access to 1.4 million PDFs of scientific publications in different domains. In the next step, the retrieved PDFs are processed using GROBID (GeneRation Of BIbliographic Data) [129], a state-of-the-art extraction engine. GROBID extracts a structured full-text representation as Text Encoding Initiative(TEI)-encoded documents, thus providing easy and reliable access paragraphs and sentences.

Entity Extraction and Typing

The entity extraction component is designed to identify and type the *domain-specific entities* contained in the fulltext of a publication. All the metadata from a paper are then published in a RDF repository, encoded according to the DMS (Dataset, Method, Software) ontology [139]. In this demo we focus on the entity types *Dataset* (i.e. dataset presented or used in a publication), and *Methods* (i.e. algorithms used to create or analyse a data set).

The Entity Extraction and Typing component is organized into two sub-components namely *NER/NET* and *Crowdsourcing*. The extraction of entities relies on NER/NETs (Named Entity Recognition/Named Entity Typing) algorithms trained with minimal human supervision. SmartPub allows the interaction with crowds for training data creation or filtering purposes.

[1] http://dblp.uni-trier.de/
[2] https://arxiv.org/

NER/NET

The goal of *NER/NET* sub-system is to address the problem of long-tail entity recognition with minimal human input. The training of *domain-specific* NER/NETs is a challenging task due to: 1) the *long-tail* nature of such entity types, both in existing knowledge bases *and* in the targeted document collections [177]; and 2) the high cost associated with the creation of hand-crafted rules or human-labeled training datasets for supervised machine learning techniques.

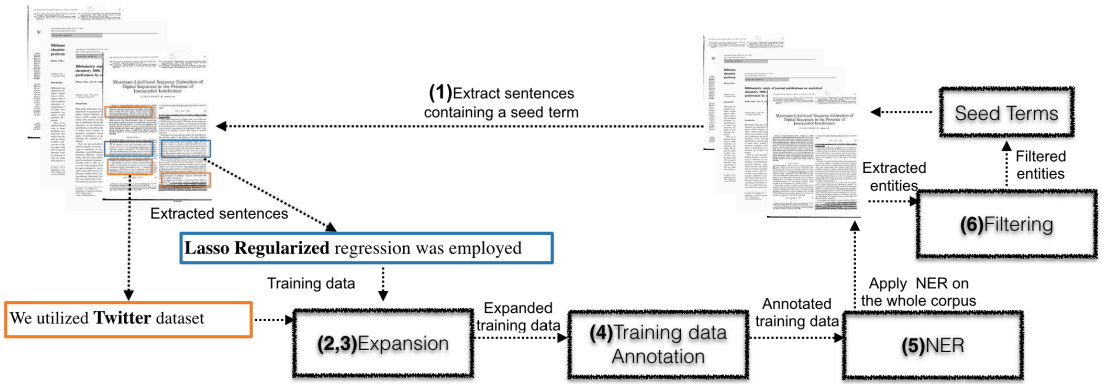


Figure A.2: Overview of the domain-specific long-tail named entities recognition approach

`SmartPub` integrates results from our previous work [143, 142], and extends them as depicted in Figure A.2. Starting from a seed set of instances of the targeted entity type (e.g. method), (1) we obtain text snippets from the publication corpus to be used in a first training data extraction step; (2) the set of seed instances are then semantically expanded to include potential yet unknown instances. For this, the word2vec model (100 dimensions) is trained on the whole corpus, as described in [153], to learn all uni- and bi-gram word vectors for all terms in the corpus. Then, we use a pre-trained entity recognition library (e.g. the one provided by the NLTK package) to obtain a list of all entities contained in the training data. Entities are then clustered with respect to their embedding vectors using K-means clustering; silhouette analysis is used to find the optimal number k of clusters. Finally, clusters that contain at least one of the seed terms are assumed to (only) contain entities of the same type. In the third step (3) the set of training snippets are semantically expanded to include sentences which are unlikely to contain instances of the desired type, but are still very similar in semantics and vocabulary to serve as informative negative examples in order to boost the NER training accuracy. For this, we rely on *doc2vec* document embeddings [111], a variant of *word2vec*, to learn vector representations of the sentences in the corpus. For each sentence in the development set, we use *doc2vec* (100 dimensions) to discover the most similar sentence which does not contain any known instance of the targeted type (i.e., expanded terms). Such sentences sometimes are likely to contain an unknown instance

of the targeted entity type, which would now be misclassified in the training set. In the fourth step (4) the training data is annotated with the expanded instance list, so to (5) train a NER that is then applied on the document corpus to extract new named entities of the given types. In a final step (6) the extracted entities are processed trough a set of filters that heuristically exclude likely misclassified instances (e.g excluding general english words using wordnet[3]), thus yielding the final result set. For training a new *NER*, we used the Stanford NER tagger[4] to train a Conditional Random Field (CRF) model.

This automatic approach relies on minimal human input (the seed set of entities), and can operate in an iterative fashion by being repeated using the result set as a seed for the next iteration. We compared our method with the BootStrapping (BS) based concept extraction approach [200], a commonly used state-of-the-art technique in scientific literature.

Experiments [143, 142] shown that our approach can provide good quality results in terms of precision/recall/fscore for the `dataset` entity type (0.77/0.30/0.43) compared to *BS* (0.08/0.13/0.10) and for the `method` entity type (0.68/0.15/0.25) compared to *BS* (0.11/0.32/0.16), with a seed set of 100 entities. We infer that different expansion strategies augment the performance of our technique compared to the *BS* which just relies on features such as unigrams, bigrams, closest verb, etc.

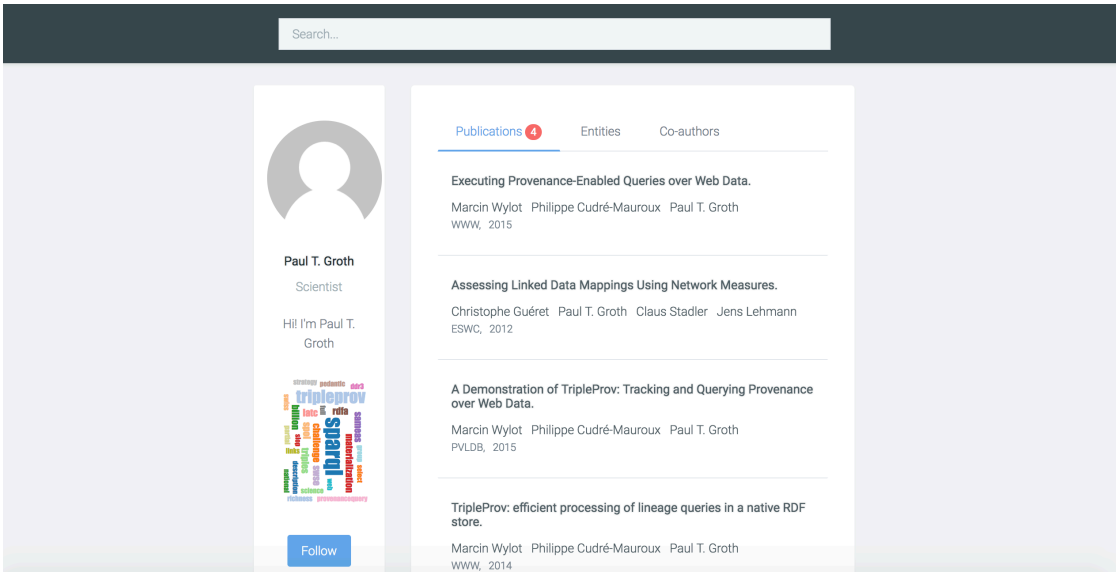


Figure A.3: Explore Authors

[3] `http://wordnet.princeton.edu/`
[4] https://github.com/dat/stanford-ner

Figure A.4: Visualize entities

Figure A.5: Examples of data visualisations dashboard of the SmartPub platform

Crowd-sourcing

The *Crowd-sourcing* [22] component is responsible to close the loop with the final users to help improving the performance of the NER/NET model. The crowd-sourcing component samples annotated sentences from the corpus and offers them the possibility to filter out irrelevant entities, so to reduce the number of false positives detected by the noisy NER. The current version of SmartPub uses the uncertainty sampling strategy[5] (e.g. least confidence, smallest margin), to rank unlabeled examples for annotation. To assess the quality of users' annotations, SmartPub currently implements a simple labeling aggregation scheme based on majority voting. Crowd-labeled sentences are then used to re-train the existing model, to achieve higher accuracy and/or identify new entity types. Moreover, the crowd-sourcing component also generates entity linking tasks. The task requires linking entities to an instance in the knowledge base, which entails annotating an ambiguous entity mention (e.g. SVM) with a link to the unique instance (e.g Support Vector Machine).

Exploration and Visualization

All the documents as well as the extracted entities in the corpus are indexed using Elasticsearch[6]. We designed an easy to use user interface to explore publications, authors and the domain specific entities (as in Figure A.5).

[5] https://github.com/ntucllab/libact

[6] https://www.elastic.co/

The publications can be explored using the title, authors name or the fulltext. For each publication, SmartPub shows the entities extracted from the fulltext. Figure A.3 shows an example of exploring authors. For each author we show, the list of publications in our corpus, list of co-authors as well as the extracted entities from the full text of the authors publications.

SmartPub currently offers the following set of visualizations for a given entity: *1) Popularity Over Time* in the shape of a stream graph. As depicted in Figure A.4 on the right, the Stream Graph displays the contribution of a given entity and its top six co-occurred entities in a certain year by means of the number of entity-occurrence. The thickness of the graph shows the popularity of the entity in a year. Stream Graphs are ideal for discovering trends over time across a wide range of categories. Different colors in the graph are indicators of different entities and the name of the entities are displayed with hover interactivity. The Stream Graph can be further filtered according to the conference using a multi-select dropdown list. *2) Popularity Over Conferences* given conference in the shape of a pie chart. Figure A.4 left shows the number of papers including a given *entity* in different conference series. 3) *Co-occured* entities in the shape of word cloud. Figure A.4 left shows the word cloud, a graphical representation of the frequency of co-occured entities.

A.3 Demo Highlights

We will present the demo using sample of scientific publications with a focus on data science and processing. In our corpus, we have 11,589 papers from ten conference series. The Joint conference on Digital Libraries (JCDL – 1,416 papers, 2001–2016); the International Conference on Theory and Practice of Digital Libraries (TPDL – 276 papers, 2011–2016); the International Conference on Research and Development in Information Retrieval (SIGIR – 412 papers, 1971–2016); the Text Retrieval Conference (TREC – 1,444 papers, 1999–2015); the European Conference on Research and Advanced Technology on Digital Libraries (ECDL – 820 papers, 1997–2010); the International Conference on Software Engineering (ICSE – 1834 papers, 1976–2016); the Extended Semantic Web Conference (ESWC – 626 papers, 2005–2016); the International Conference On Web and Social Media (ICWSM – 810 papers, 2007–2016); the International Conference on Very Large Databases (VLDB – 1884 papers, 1975–2007); and the International World Wide Web Conference (The Web Conference – 2067 papers, 2001–2016). The demonstration will focus on exploring scientific papers, authors as well as visualizing entities extracted from the full text of the publications by means of their popularity over time or conferences.

The demonstration starts by searching for publications containing an entity name (e.g. clueweb). A list of relevant publications is listed, showing meta data such as the title,authors name as well as the venue and publication year. By clicking on the author name, we can navigate to the *author* page. For each author, SmartPub shows publications in the corpus, the list co-authors, and the list of entities extracted from the author's publications, which are shown as a word cloud below the name of the author. Entities

in the word cloud are clickable, leading to a separate tab called *Entities* which contains the list of entities with their corresponding entity types.

By clicking on each of the publications title we can navigate to the *publication* page which contains the abstract, the references as well as the entities extracted from the full text of the papers. By clicking on each of the entities listed in the entity tab we navigate to the *entity* page. For each entity, `SmartPub` offers a set of visualizations described in Section A.2. As an example for the entity name `Clueweb`, in the stream graph we show the popularity of `Clueweb` and its top six co-occured entities (i.e. `wikipedia`, `urls`, `trec`, `nist`, `dbpedia`, `bm25`) in a certain year which can further be filtered based on a given conference. The Pie chart on the left shows that the `Clueweb` entity is mostly popular in information retrieval conferences such as TREC and SIGIR. The word cloud below the entity name depicts the co-occured entities with the given entity, which are all clickable. The users are able to search for any entity using the search box below the entity name on the left. Finally an example of a crowdsourcing task is shown, where the users are asked to select the appropriate label for the highlighted token.

Acknowledgments. This research has been supported in part by the Dutch national e-infrastructure with the support of SURF Cooperative (Grant Agreement No. e-infra170126).

Appendix B

LOREM: Language-consistent Open Relation Extraction from Unstructured Text

This appendix tackles the problem of multilingual open relation extraction when limited training data is available.

Abstract

We introduce a Language-consistent multi-lingual Open Relation Extraction Model (LOREM) for finding relation tuples of any type between entities in unstructured texts. LOREM does not rely on language-specific knowledge or external NLP tools such as translators or PoS-taggers, and exploits information and structures that are consistent over different languages. This allows our model to be easily extended with only limited training efforts to new languages, but also provides a boost to performance for a given single language. An extensive evaluation performed on 5 languages shows that LOREM outperforms state-of-the-art mono-lingual and cross-lingual open relation extractors. Moreover, experiments on languages with no or only little training data indicate that LOREM generalizes to other languages than the languages that it is trained on.

B.1 Introduction

Extracting relationships between entities from text is a core building block for (semi-)automatically creating structured knowledge bases. Relation extractors focusing on lexical features and smaller sets of relationship types have shown to be effective, especially in defined domains like bio-medical [64, 173] or law. However, they struggle in less focused applications like general-purpose Web or Social Media mining which are not restricted in relation type or language used. In this paper, we target this use case with a novel open relation extraction model which is also coping with multi-linguality.

Open Relation Extraction (ORE) is defined as the process of discovering arbitrary semantic connections between entities in unstructured texts [43]. Given an input sentence

such as *"Turing was born in England in 1912"* and two entities like *<Turing, England>*, an ORE system should extract a sub-string which entails the semantic relation between the two entities (i.e. *"was born in"*).

Initially, ORE research focused on training sequence tagging models by utilizing external NLP tools (such as POS taggers) and manually defined lexical and syntactic features [60, 135, 48, 8]. The dependency on external NLP tools results in error propagation. Also, most of these tools are developed for English only hindering the adoption of ORE algorithms to other languages. Although being a rough estimate, various cross-over studies imply that around 70% of the internet is written in languages other than English [209]. This indicates a need for more generic, language-agnostic ORE models. Recent approaches [222, 42, 199, 95] employed deep neural networks to automatically learn relation patterns from large training sets to tackle the problem of manually defining features and language structures for multiple languages. However, they still require additional NLP tools for pre-processing text such as translators or dependency parsers, thus limiting easy extension to new languages.

Our goal is to exploit similarities and pattern consistencies which exist between many natural languages to replace those language-specific external tools. Recently, Relaxed Cross-domain Similarity Local Scaling (RCSLS) [97] was presented, a word embedding alignment approach which exploits the inter-dependencies between any two languages and maps all monolingual embeddings into a shared multilingual embedding space. In a similar fashion, we leverage existing pre-trained multilingual word embeddings (which are currently available for 44 languages[1]). The intuition behind these efforts is that some languages share common ancestry, and thus exhibit similarities in grammar and vocabulary. We therefore assume that also their trained relation extractors can support each other, which is especially valuable for use cases where a well-trained model is available, but relation extraction is required for a resource-scarce language. For example, we can show that a richly trained English relation extraction model (for which many manually annotated training corpora are available) can significantly boost the performance of a poorly trained Dutch model (for which only very few training samples are available.)

Based on this intuition, we present LOREM, a model that harvests information that is consistent over languages for Open Relation Extraction. LOREM depends only on monolingual ORE training data and multilingual word embeddings, it can thus be easily extended to new languages.

We make the following contributions:

- We introduce a Language-consistent Open Relation Extraction Model (LOREM). To the best of our knowledge, LOREM is the first open relation extractor that utilizes language-consistent relation structures to improve open relation extraction performance across multiple languages. In addition LOREM does not depend on language-specific knowledge or external NLP tools such as translators or dependency parsers, thus allowing for easy expansion to new languages.

[1] https://fasttext.cc/

- To the best of our knowledge, we are the first to employ multilingual, aligned word embeddings as the input of a multilingual relation extractor. Our experiments show that this improves the performance over using conventional monolingual word embeddings.
- We present experimental results on five high-resource languages showing that LOREM outperforms state-of-the-art mono-lingual and cross-lingual open relation extractors. Additionally we present experiments on no- and low-resource languages which demonstrate the ease and effectiveness of expanding LOREM to additional languages. This shows that language consistency can not only boost extraction performance for low-resource languages (like Dutch which can benefit from English), but also high-resource languages (like a well-trained English model which still benefits slightly from e.g. a French one),

We removed references to our source code to maintain anonymity, but upon acceptance we will make our source code available to the community.

B.2 Related Works

From the literature, we identify two paradigms; *closed* and *open* relation extraction. Within the closed paradigm, the goal is to classify a sentence as being part of a predefined set of relation classes. Banko et al. [13] argue that requiring pre-defined relation classes is too limiting for many real-world applications. To alleviate this requirement, they propose the open relation extraction (ORE) paradigm. The vast majority of ORE research is presented for the English language. Although multilingual methods were proposed, they either depend on bilingual training data or solely work in the closed relation extraction domain.

English Open Relation Extraction

EORE was first introduced by Banko et al. [13]. Conventional models use lexical and syntactic features that rely on external NLP tools and language-specific relation structures. To avoid error propagation by these external tools and alleviate the burden of designing manual features, multiple neural open relation extractors were proposed [42, 199, 95]. Jia et al. [95] present the current state-of-the-art model called NST. They define a tagging scheme and predict a tag for each word in the input sentence. For this purpose, they jointly train a CNN and bi-LSTM. The output of these models is fed into a final CRF layer to end up with the final prediction. Their experiments show that CNNs and LSTMs provide complementary information for the RE task.

Even though recent research efforts yield state-of-the-art results for the ORE task by utilizing neural network based models, these works are solely focused on the English language and will encounter two weaknesses when applied in a multilingual setting. First, the vast majority of these systems use external NLP tools such as PoS-taggers and dependency parsers [60, 135, 48, 8] and need to be adapted to use tools for the given

language, which is a non-trivial process. Second, EORE would fail to exploit information that is present over multiple languages (language-consistent patterns). Both of these weaknesses are addressed by two different multilingual RE techniques; cross-lingual RE and language-consistent RE. Cross-lingual systems try to extract relations from a source language by exploiting information and systems from a target language, thereby removing the need for a labelled training set or NLP tools in the source language. On the other hand, language-consistent systems exploit information that is present in multiple languages.

Cross-lingual Open Relation Extraction

Cross-lingual approaches can be used when we need to extract relations from a source language for which we do not have a labelled training set. We do however need to possess either a performant translator [61] or a sufficiently large bi-text corpus between English and the source language [223]. In recent years multiple cross-lingual approaches are introduced [61, 222, 223]. Typically, a cross-lingual system translates the source language into the target language (e.g. English) and employs an existing relation extractor. In an effort to relax the translator assumption and to tailor the translator to the RE task at hand, Zhang et al. [222] present their joint Machine Translation/Information Extraction (MT/IE) system. Instead of first translating the source text and then applying a relation extractor, they jointly train a machine translation and relation extraction model. The translator assumption is replaced by the assumption that a bi-text corpus (e.g. a corpus of Chinese sentences and their English translations) is available. Notwithstanding the fact that cross-lingual relation extractors can be used to extract relations from multiple languages, they require additional bilingual data and fail to explicitly exploit language-consistent information. In addition they only exploit information from the target language, disregarding patterns that might be specific to the source language or patterns that are consistent over languages.

Language-consistent Relation Extraction

To remove the dependency on bi-text corpora or translators and to introduce a mechanism for exploiting information that is consistent over languages, language-consistent relation extraction was proposed. Language-consistent RE literature [123] assumes that relation patterns in sentences are substantially consistent between different languages. This assumption can be exploited to train a single model which gathers information from multiple languages. In the previous section, we have seen that cross-lingual relation extractors exist for the open RE domain. In contrast, language-consistent relation extractors are currently solely proposed for the closed domain [123, 212].

Wang et al. [212] train a separate language-individual model for every language and one language-consistent model on all languages in the closed RE paradigm. By combining both models, they can utilize relation patterns that are specific to languages as well as patterns that are consistent over languages. To ensure that the representations of sentences are aligned over multiple languages, they use an adverserial training approach.

To obtain the same latent consistency among languages in similar NLP tasks, multilingual word embeddings were proposed [97, 31] which are trained with the specific goal to align similar words over multiple languages. Multilingual word embeddings use information from multiple high resource languages to create a shared embedding space in which also low-resource language can be represented.

Our work is inspired by Jia et al. [95] and Wang et al. [212]. In contrast, LOREM utilizes language-consistent information within the Open Relation Extraction domain and employs multilingual embeddings [97] for multilingual relation extraction. Our approach does not rely on any external NLP tools or additional bilingual training data, ensuring low-cost extendibility to new languages.

B.3 LOREM: Language-consistent Open Relation Extraction Model

In a nutshell, the idea behind our Language-consistent Open Relation Extraction Model is to start with several language-individual models for each required language. This means that for each language, at least some training data needs to be available (however, as we can show in our experiments for the Dutch language, it can be sufficient to have only a few hundred training samples available which one of the authors could easily provide by himself.) In Figure B.1, one language-individual model (to the left) is depicted, but to take full advantage of LOREM, several of such models should be available. We base these models on Neural Sequence Tagging (NST) [95], a recent state-of-the-art approach for (mono-lingual) open relation extraction.

To exploit consistencies between the languages available to the system, we additionally train a language-consistent model using all available languages. The techniques for combining the individual models and the language-consistent model is inspired by AMNRE [212] (a model for language-consistent relation extraction, which is strictly limited to a closed domain of few relationship types. However, we changed the workflow of AMNRE considerably to work with the NST models, e.g. by switching to multilingual embeddings). In the current version of LOREM, only one language-consistent model is used. But as discussed in the conclusions, we see potential for having several of those models which are trained on selected subsets of the available languages.

Input Embeddings

An input sentence is encoded using two different types of pre-trained word embeddings, one for use with the language-consistent model and one for use with the language-individual models. For the language-individual models, we use conventional pre-trained word embeddings. In Figure B.1, these embeddings are represented in blue on the left. The training sentences of the language-individual model all come from the same language, and we expect that this model finds relation structures that are specific to that individual language.

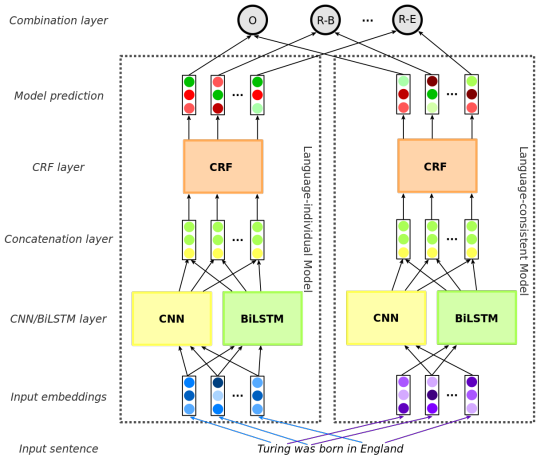


Figure B.1: Architecture of our Language-consistent Open Relation Extraction Model (LOREM).

In order to achieve latent consistency among languages, we pioneer the use of multilingual embeddings for the language-consistent model [97]. By using embeddings that are aligned over languages, we hypothesize that we can ease the burden of the CNN/-BiLSTM layer to extract language-consistent patterns. Here, the intuition is that the multi-lingual embedding prevents language-specific clusters in the embedding space (such clusters naturally happen when using multiple mono-lingual embeddings). Thus, related or similar words should be close no matter their original language which supports discovery of language-consistent patterns. Note that we use pre-trained embeddings in our current version of LOREM. In scenarios where such dependencies are undesired, such embeddings could also be custom-learned during system setup.

In Figure B.1, these embeddings are represented in purple on the right. For this model, the training sentences come from multiple languages. Thus, we expect this model to extract relation patterns consistent over these language.

In addition to word embeddings, entity tag vectors are added to the input. These are simple one-hot encoded vectors which indicate if the current word is part of the first, second or no relation entity. Please note that in contrast to the NST model, we do not use Part-of-Speech tags since these introduce a dependency on PoS-taggers.

The input sentence is represented as a k-dimensional embedding sequence $\mathbf{x} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n\}$, where $\mathbf{w}_t$ is the representation of the t^{th} word of an input sentence that has n words. Here, $k = k_i + k_c$, k_i and k_c are the dimensonalities of the language-individual and -consistent model input respectively. $k_i = k_{mono} + k_e$ and $k_c = k_{multi} + k_e$, where k_{mono} is the dimensionality of the monolingual word embedding, k_{multi} of the multilingual word embedding and k_e of the entity tag vector.

NST Layers

The next four layers (CNN/BiLSTM, concatenation, CRF, model prediction) are identical to the NST model. We shortly reiterate the NST model's general architecture, a more detailed description can be found in the original NST paper [95]. Relational words tend to occur in the neighbourhoods of entities. Therefore, certain parts of the input sentence might have a higher chance of containing relation words than others. A CNN is used to capture this local feature information from the input sentence. At the same time, a bidirectional LSTM is used to capture the forward and backward context of each word, including long-distance relations. By concatenating the outputs of the CNN and the forward and backward pass of the LSTM, a continuous representation of each word in the input sentence is formed. Next, these representations are used as the input for a straightforward CRF layer, which tags a word using the NST tagging scheme.

Tag	Meaning
R-S	Single word relation sub-string.
R-B	Beginning of relation sub-string.
R-I	Inside the relation sub-string.
R-E	Ending of relation sub-string.
O	Outside the relation sub-string.

Table B.1: NST tagging as proposed by Jia et al. [95].

The NST tagging scheme consists of five possible relation tags, which can be found in Table B.1. The sentence *"Alan Turing was born in England."* should be tagged as follows; *"Alan$_O$ Turing$_O$ was$_{R-B}$ born$_{R-I}$ in$_{R-E}$ England$_O$.$_O$"*.

The output of the NST layers are two prediction sequences $\mathbf{y}_{ind} = \{\mathbf{i}_1, \mathbf{i}_2, ..., \mathbf{i}_n\}$ and $\mathbf{y}_{con} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_n\}$, where $\mathbf{y}_{ind}$ contains the predictions of the language-individual model and $\mathbf{y}_{con}$ contains the predictions of the language-consistent model. $\mathbf{i}_t$ and $\mathbf{c}_t$ are the 5-dimensional prediction vectors of the language-individual and -consistent models respectively. For the original NST model, these are binary vectors which contain a 1 for the predicted tag and a 0 for all other tags. After our alteration, these vectors contain a probability score for each of the possible relation tags. This allows us to fittingly combine the predictions of the language-individual and -consistent models in the next layer.

Combination Layer

In the last layer, we define the final probability sequence by $\mathbf{y} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ with

$$\mathbf{p}_t = \mathbf{i}_t \odot \mathbf{c}_t, \tag{B.1}$$

for the t^{th} word in the input sentence[2]. The output tag sequence is defined by $\mathbf{z} = \{z_1, z_2, ..., z_n\}$ where

$$z_t = \arg\max_j \mathbf{p}_{tj} \tag{B.2}$$

[2] $\odot$ is used as the Hadamard product.

and where $\mathbf{p}_{tj}$ is the j^{th} element of $\mathbf{p}_t$.

LOREM might (rarely) yield tag sequences which are invalid. This is a common issue with sequence taggers, including also vanilla NST. For example, the tag for a single word relation (*R-S*) can not be followed by a tag for the end of a multi-word relation (*R-E*). In this case, the first tag could be changed to *R-B* to form a valid tag sequence. We create two different versions of LOREM, LOREM$_{clean}$ which alters invalid sequences to valid sequences and LOREM which allows invalid sequences. To create LOREM$_{clean}$, we transfer the predicted tags to binary tags (*R* if the word is in the relation, *O* if it is not). Next, we specify the *R* tags so that the first *R* occurrence in a sentence will become *R-B* for a multi-word relation and *R-S* for a single-word relation. Similarly, the last *R* occurrence will become *R-E* and the middle *R* occurrences will become *R-I* for a multi-word relation. Please note that this approach solely influences the specific relation tag that is given to a word, it does not influence whether a word is tagged as being part of the relation or not.

B.4 Experiments

We present experimental results investigating the behaviour of LOREM and its sub-models guided by the following hypotheses:

H1: For *high*-resource (i.e. 100k+ sentences with tagged open relations) languages, LOREM outperforms state-of-the-art monolingual open relation extractors (including NST) by additionally harvesting language-consistent relation patterns from multilingual texts.

H2: Multilingual word embeddings improve the performance of the language-consistent sub-model, and thereby the performance of LOREM by introducing a latent consistency among languages.

H3: For *low*-resource (in our case ∼750 tagged sentences) and *no*-resource (i.e. no sentences with tagged open relations) languages, our approach is able to outperform language-individual models by harvesting language-consistent relation patterns from multilingual texts and by utilizing models of languages that have a similar origin.

Our model uses the hyper-parameters that were proposed by Jia et al. [95] for their NST model. We evaluate the performance of our approach using precision, recall and F_1-score.

Datasets

Information about the used training and test data is presented in Table B.2. We used data from the following datasets, covering English, Spanish, French, Hindi, Russian,

	High					No	Low
	English	Spanish	French	Hindi	Russian	Italian	Dutch
# Training sentences	576,462	429,413	468,625	280,815	550,720	0	750
# Test sentences	2,191	246	512	622	573	10,000	100
Origin training data	NeuralOIE	WMORC$_{auto}$	WMORC$_{auto}$	WMORC$_{auto}$	WMORC$_{auto}$	-	WMORC$_{auto}$
Origin test data	ClausIE	RWP	WMORC$_{human}$	WMORC$_{human}$	WMORC$_{human}$	WMORC$_{auto}$	MC

Table B.2: Description of the datasets used in our experiments for high-, no- and low-resource languages. Legend: *RWP* – Raw Web/Parallel En-Sp; *MC* – Manually Created

Italian, and Dutch.[3]

WMORC [61] WMORC contains manually annotated open relation extraction data for 3 languages (WMORC$_{human}$) and automatically tagged (and thus less reliable) relation data for 61 languages, created using a cross-lingual projection approach (WMORC$_{auto}$). The sentences are gathered from Wikipedia.

NeuralOIE [42] English dataset created by using only high-confidence extractions of an existing relation extractor [134] from Wikipedia sentences.

ClausIE [48] Three manually annotated English test sets from Wikipedia and New York Times sentences.

Raw Web/Parallel En-Sp [225, 226] Two manually annotated Spanish test sets from school text book and web page sentences.

Custom For Dutch, we created our own test set by having a native speaker tag 100 random Dutch Wikipedia sentences (since the Dutch sentences contained in WMORC$_{auto}$ seemed to be of too low quality to be used for testing due to their automatically generated nature).

The size of our high-resource training sets (En, Sp, Fr, Hi, Ru) is comparable to the dataset used in the original NST paper [95]. Moreover, early tests did not show substantial benefits of adding more data after this point. We approach Dutch from a low-resource scenario, so we only sample 750 Dutch sentences from WMORC$_{auto}$ for training. We don't use any Italian training data, since Italian is used as a no-resource language in our experiments (i.e. for Italian, there is no language-indiviudal NST model available during the evaluations, only the language-consistent one). For training the language-consistent model, we sample the high-resource datasets presented in Table B.2, so that the combined set of all five languages contains 450,000 - 550,000 training sentences. The selected samples are balanced across these languages. This way, we can make a fair comparison between the language-individual and -consistent sub-models since they are trained on the same amount of training data.

Given the very limited scope of existing multilingual open relation extraction literature, there are only very few results presented for these datasets ('Origin test data' in B.2) . Moreover, these were the only publicly available ORE test sets we could find for

[3]We selected these languages, since these are the only languages for which we could find openly available test data.

non-English languages. The Italian test set is created by sampling 10,000 sentences from WMORC_{auto}. Since these sentences are automatically tagged, we do expect a higher noise level than in the manually tagged test sets.

For the language-individual model, we use FastText word embeddings [74] which are trained on Common Crawl and Wikipedia dataset. For the language-consistent model, we use pre-trained multilingual embeddings which are released by FastText [97] for 44 languages. The vectors were trained on a Wikipedia dataset. The dimensionality of both embeddings is 300.

Comparison Methods

During our experiments, we compare LOREM to a range of previously proposed methods. For English, we compare LOREM to the same baseline systems that were used during the evaluation of the NST model by Jia et al. [95]. These include:

NST_{no-PoS} [95] The NST model forms the underlying model of LOREM, yet there are differences between the two. The original NST model does not contain a language-consistent part. We present the results for NST without PoS-tags for a fair comparison.

Reverb [60] Reverb exploits syntactic and lexical constraints on binary relations expressed by verbs.

OLLIE [135] This model designs complex patterns using syntactic processing (e.g. dependency parsers).

ClausIE [48] ClausIE exploits linguistic knowledge about English grammar to detect and identify clauses and their grammatical function.

Open IE-4.x [134] This is a combination of a rule-based Open IE system and a system which analyzes the hierarchical structure between semantic frames to construct multi-verb open relation phrases.

For Spanish, we compare LOREM to;

ExtrHech [226] A system based on syntactic constraints over PoS-tag sequences targeted at Spanish.

ArgOE [66] ArgOE uses dependency parsers to extract a set of propositions for different argument structures.

Finally, we compare LOREM to a **cross-lingual system** presented by Faruqui et al. [61] which utilizes a translator and an English ORE system.

	English			Spanish			French			Hindi			Russian		
Model	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Our work															
LOREM	.801	.757	**.782**	.615	.522	.564	.783	.715	**.747**	**.900**	.598	**.719**	**.762**	.719	.740
LOREM_{clean}	.782	**.765**	.774	.585	.547	.564	.726	**.729**	.727	.687	**.618**	.651	.709	.726	.718
Language-ind.	.796	.747	.771	.595	.498	.541	.781	.693	.735	.878	.540	.667	.755	**.741**	**.748**
Language-con.	.792	.734	.762	.583	.471	.521	.733	.673	.702	.813	.566	.667	.712	.690	.701
English															
NST_{no-PoS}	.783	.708	.744	-	-	-	-	-	-	-	-	-	-	-	-
Reverb	.641	.162	.259	-	-	-	-	-	-	-	-	-	-	-	-
OLLIE	**.985**	.242	.389	-	-	-	-	-	-	-	-	-	-	-	-
ClausIE	.801	.531	.638	-	-	-	-	-	-	-	-	-	-	-	-
Open IE-4.x	.792	.331	.467	-	-	-	-	-	-	-	-	-	-	-	-
Spanish															
ExtrHech	-	-	-	**0.710**	**0.595**	**0.647**	-	-	-	-	-	-	-	-	-
ArgOE	-	-	-	0.500	-	-	-	-	-	-	-	-	-	-	-
Cross-lingual															
Faruqui et al.	-	-	-	-	-	-	**0.816**	-	-	0.649	-	-	0.635	-	-

Table B.3: Results of LOREM, its sub-models and existing models. Bolds indicate the best values per language.

B.5 Experimental Results

H1: LOREM for High-resource Languages

Table B.3 contains the experimental results of LOREM and the comparison methods on five different high-resource test languages. We find that both LOREM models outperform all English baseline systems in terms of recall and F_1-scores. Focusing on the comparison with the NST model, we find that LOREM outperforms NST on precision, recall and therefore F_1-score. The high F_1-scores of our LOREM models are mainly due to the excellent recall scores, compared to other systems. LOREM achieves the best presented F_1-score on the ClausIE datasets when PoS-tags are not used.

Next, we compare LOREM to two Spanish open relation extractors. It is important to note that both existing models heavily rely on semantic constraints and external NLP tools. For ArgOE the authors only present a precision score. The results show that LOREM is outperformed by ExtrHech on the Spanish datasets. It does however achieve a higher precision than ArgOE. Even though the evaluation results are not quite as high as the current state-of-the-art model, LOREM does have the big advantage that a user does not have to manually define semantic constrains.

We now turn our attention towards the three remaining test languages. To the best of our knowledge, there exists only one system for which results are published on the WMORC_{human} test set, being the cross-lingual model by Faruqui et al. [61]. For this model, the source code is not available and only precision scores are presented. We find that the cross-lingual model slightly outperforms LOREM in terms of the French precision score. However, LOREM clearly outperforms the cross-lingual model on both Hindi and Russian. This might be caused by the fact that the cross-lingual model is heavily dependent on a translator from English to the target language and an existing English relation extractor. LOREM eliminates this dependency by introducing a

language-consistent component. The results indicate that this improves the generalizing capabilities over languages, providing prove for the validity of hypothesis 1.

In order to investigate how well each submodel in LOREM performs, we presented the results obtained by the sub-models in Table B.3. LOREM generally outperforms both the language-individual and -consistent model, showing the merit of combining these sub-models. This falls in line with the conclusions presented by Wang et al. [212] for the closed domain.

In addition to these findings, we also observe a returning pattern between LOREM and LOREM$_{clean}$. For all languages, LOREM achieves higher precision and F_1-scores, indicating a better overall performance. However, cleaning the prediction results does consistently improve the recall of the model. Thus, we conclude that LOREM generally outperforms LOREM$_{clean}$, yet LOREM$_{clean}$ should be used when recall is crucial for the application domain.

Another, somewhat surprising, observation from Table B.3 is the reasonably good performance of the language-consistent model, given the fact that this sub-model is not trained on one specific language. From these results, we wondered if relation structures truly differ a lot between languages. It could be the case that a language-individual model already performs reasonably well on other languages, eliminating the need for a language-consistent model. To test this hypothesis, we compare the average results of the language-consistent model over all five languages to the average results of the language-individual models on these languages. The results are presented in Table B.4. The results clearly counteract the hypothesis, showing the merit of a language-consistent model over simply using one language-individual model for every language.

Model	**P**	**R**	**F$_1$**
Language-consistent	**.727**	**.627**	**.671**
English language-individual	.393	.317	.347
Spanish language-individual	.586	.390	.455
French language-individual	.679	.464	.543
Hindi language-individual	.266	.110	.138
Russian language-individual	.632	.483	.546

Table B.4: The average prediction results of the language-consistent model and language-individual models on all test languages. The bolds indicate the best values.

H2: Multi VS Monolingual Embedding

Current multilingual relation extraction literature utilizes monolingual word embeddings to encode sentences of different languages. However, we expect the model to extract patterns that are consistent over languages. Therefore, the model should ignore the language in which an input word is written. Naturally, aligning these word embeddings of languages would ease the burden of the language-consistent model to extract language-consistent relation patterns.

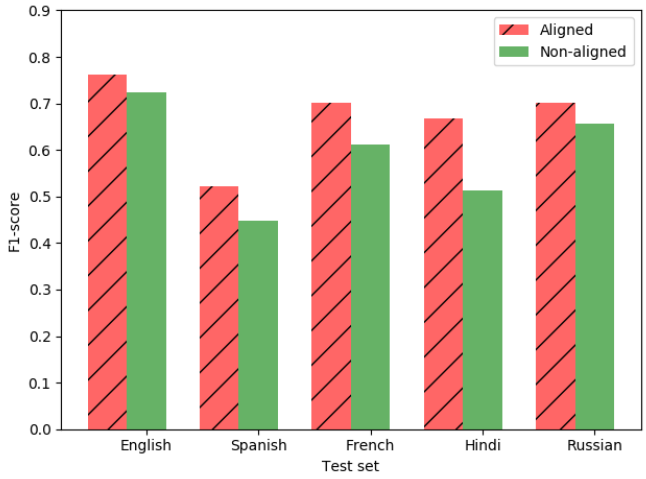


Figure B.2: F_1-score using aligned and non-aligned embeddings for the language-consistent sub-model.

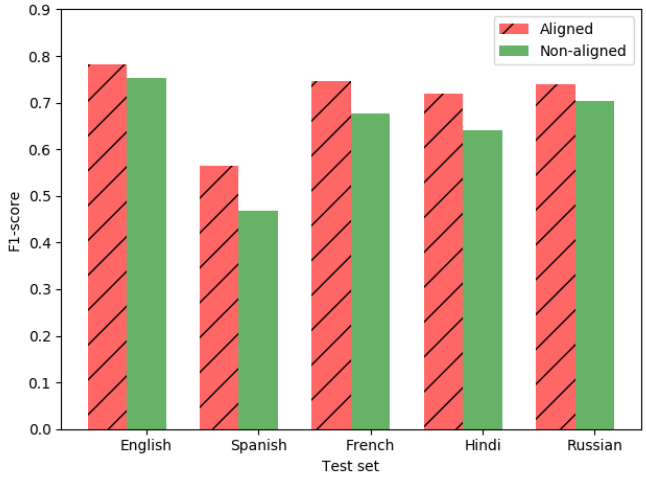


Figure B.3: F_1-score using aligned and non-aligned embeddings for LOREM.

To examine this hypothesis, we compare the results obtained by using both non-aligned (monolingual) and aligned (multilingual) word embeddings. In Figure B.2 and B.3, we present the results of this experiment. Additionally, we provide the impact of both approaches on the full LOREM model, showing that improvements for the language-consistent sub-model indeed lead to improvements of the full model. We observe that the aligned word embeddings yield better performance on every language for both the language-consistent sub-model and the full LOREM model in terms of F_1-score. Given these test results, we can confirm the validity of hypothesis 2.

H3: LOREM for Low/No-resource Languages

The evaluation results for low- and no-resource languages are shown in Table B.5 and B.6. If no open relation extraction training data is available for a certain language, our model can still be utilized in three possible ways: 1) we can use the language-consistent sub-model trained on other languages, 2) we can use a language-individual model of a language that has a similar origin to the current language 3) or we can combine both into a full LOREM model. If we also have a very small training set of around 750 sentences (for the low-resource scenario), we can additionally train a language-individual model using it.

	Dutch			**Italian**		
Model	P	R	F_1	P	R	F_1
Language-con.	.705	**.633**	.667	.506	**.342**	**.408**
English	.655	.582	.616	.293	.232	.259
Spanish	.441	.306	.361	.435	.203	.277
French	.685	.510	.585	.352	.217	.268
Hindi	.000	.000	.000	.362	.029	.054
Russian	.703	.265	.385	.393	.164	.232
LOREM	**.744**	.622	**.678**	**.554**	.246	.341
LOREM_{clean}	.663	.622	.642	.383	.287	.328

Table B.5: (no-resource) Results of the language-consistent model, language-individual models and LOREM on the Dutch and Italian test sets.

Model	**P**	**R**	**F_1**
Language-individual	**.786**	.444	.568
LOREM	.753	**.646**	**.696**

Table B.6: (low-resource) Results of low-resource models on the Dutch test set.

For the no-resource scenario, Table B.5 provides the results for Dutch and Italian test sets. We hypothesize that language-individual models of languages that have a similar origin as the test language will yield better results than those of languages with a different origin. If we focus on the F_1-scores, we find a general pattern that adheres to this intuition. For the Dutch test set, the English model yields the highest F_1-score. This is to be expected since English and Dutch are the only two West-Germanic languages in this experiment. The French model also performs reasonably well, this can be explained by the fact that French and Dutch are both of European origin. Given that French and Spanish are both Romance languages, we would expect similar results on the Dutch test set. Yet, the Spanish model performs significantly worse and does therefore not follow our intuition. The Russian model also yields worse results than the French and English models. This can be explained by the fact that Russian has a Slavic origin. The Hindi model on the other hand is not able to find any valid relations. Given that all other

languages have a European nature and Hindi has an Indo-Iranian nature, this behaviour falls in line with our intuition. A similar pattern can be observed for the Italian test set, albeit less distinct.

For both Dutch and Italian, the language-consistent model outperforms all language-individual models. This shows the merit of combining languages to find language-consistent relation patterns. For the Dutch test set, LOREM even further improves the F_1-score. This is not the case for the Italian test set. These experiments show the first application of an open relation extractor on a different language than it was trained on without the need for a translator. More experiments on different test sets are needed to derive solid conclusions on the matter. Yet, our experiments provide a first indication of the validity of hypothesis 3.

For the low-resource scenario, if we compare the results shown in the top entry of Table B.6 to the evaluation results presented in Table B.5, we find that the low-resource Dutch language-individual model is outperformed by the English language-individual model. This indicates that a high-resource model in a similar language outperforms a low-resource model in the test language. However, since we now have a Dutch language-individual model, we can combine it with the language-consistent model to form a full LOREM model. Comparing these results to Table B.5, we see that the LOREM model that employs the Dutch language-individual model outperforms all models from the no-resource scenario. This is another indication of the validity of hypothesis 3 for the Dutch test set. Again, more experiments need to be conducted to derive more general conclusions.

Until now, we trained a full language-individual model for the low-resource language, ignoring the fact that we might need to treat a low-resource scenario differently than a high-resource scenario. It is a well-known phenomenon that more complex models generally require more training data, since more parameters need to be optimized. We have examined the possibility of only using a CNN or Bi-LSTM instead of both, to reduce the number of parameters. Results show that although LOREM$_{LSTM}$ and LOREM$_{CNN}$ achieve a higher precision scores than LOREM (0.802 and 0.836 to 0.753), this comes at the expense of a lower recall scores (0.616 and 0.566 to 0.646). As a result, the F_1-scores of are lower than or equal to those of LOREM (0.696 and 0.675 to 0.696). Therefore, we did not find a clear advantage of simplifying the model in this low-resource scenario. Please note that results presented by Jia et al. [95] clearly show that combining a CNN and LSTM outperforms both separate models for the high-resource ORE task.

Qualitative Analysis

Next to the quantitative analysis, we also conducted a qualitative analysis on the English test sets.

True positives: We found that LOREM is better at extracting relations that follow abnormal patterns than the language-individual sub-model. For example, given the sentence *"The market wants to do better, said Gregory Bundy, head of equity trad-*

ing." and entity tuple `<Gregory Bundy, The market wants to do better>`, the language-individual model does not find a relation, while LOREM extracts `said` as being the relation. Here, we find that the language-consistent component provides additional information which allows relations to be extracted, even if the entities appear in reverse order. It is likely that such patterns occur in multiple languages from which LOREM learned them, even if they were not present in the English training set. Such examples illustrate the benefits of LOREM over a language-individual approach.

False Positives and Negatives: Upon manual inspection, we find that the majority of errors arise from relations that contain multiple words. In these cases, LOREM extracts either too many or too few words compared to the ground truth relations. Typical examples include *"BIC is being sued by people who say their lighters exploded."* and *"The region is still far from rebuilt."*, from which LOREM extracts `is being sued` and `is still`, while the ground truth values are `is being sued by` and `is` respectively. These examples show that although the extraction is not completely correct, the relation is still captured to a certain extent in many cases. The test set also contains sentences from which LOREM can not extract any relations. A typical error occurs when we want to extract relations that occur between more than two entities. Given a sentence like *"28 Square miles of antennae and computers that message smart fridges, robot lawn mowers and smart doorbells vacuum up satellite and radio communications."* with entity tuple `<28 Square miles of antennae, radio communications>`, LOREM finds no relations even though the relation `vacuum up` is present between multiple entities in this sentence.

B.6 Conclusions and Future Work

In this work, we have presented a Language-consistent Open Relation Extraction Model; LOREM. The core idea is to augment individual open relation extraction mono-lingual models with an additional language-consistent model representing relation patterns shared between languages. Our quantitative and qualitative experiments indicate that harvesting and including such language-consistent patters improves extraction performances considerably while not relying on any manually-created language-specific external knowledge or NLP tools. Initial experiments show that this effect is particularly valuable when extending to new languages for which no or only little training data is available. In these cases, LOREM and its sub-models can still be used to extract valid relationships by exploiting language consistent relation patterns. As a result, it is relatively easy to extend LOREM to new languages as providing only some training data can be sufficient. However, evaluating with additional languages would be required to better understand or quantify this effect.

Additionally, we conclude that multilingual word embeddings provide an effective approach to introduce latent consistency among input languages, which proved to be beneficial to the performance.

We see many opportunities for future research within this promising domain. More improvements could be made to the CNN and RNN by including more techniques proposed in the closed RE paradigm, such as piecewise max-pooling [220] or varying CNN

window sizes [161]. An in-depth analysis of the different layers of these models could shine a better light on which relation patterns are actually learned by the model.

Beyond tuning the architecture of the individual models, enhancements can be made with respect to the language consistent model. In our current prototype, a single language-consistent model is trained and used in concert with the mono-lingual models we had available. However, natural languages developed historically as language families which can be organized along a language tree (for example, Dutch shares many similarities with both English and German, but of course is more distant to Japanese). Thus, an improved version of LOREM should have multiple language-consistent models for subsets of available languages which indeed posses consistency between them. As a starting point, these could be implemented mirroring the language families identified in linguistic literature, but a more promising approach would be to learn which languages can be effectively combined for boosting extraction performance. Unfortunately, such research is severely hampered by the lack of comparable and reliable publicly available training and especially test datasets for a larger number of languages (note that while the WMORC_auto corpus which we also use covers many languages, it is not sufficiently reliable for this task as it has been automatically generated). This lack of available training and test data also cut short the evaluations of our current variant of LOREM presented in this work.

Lastly, given the general set-up of LOREM as a sequence tagging model, we wonder if the model could also be applied to similar language sequence tagging tasks, such as named entity recognition. Thus, the applicability of LOREM to related sequence tasks could be an interesting direction for future work.

Appendix C

Give it a shot: Few-shot learning to normalize ADR mentions in Social Media posts

This appendix tackles the problem of normalizing long-tail entities for which low number of training samples are available.

Abstract

This paper describes the system that team myTomorrows-TU Delft developed for the 2019 Social Media Mining for Health Applications (SMM4H) Shared Task 3, for the end-to-end normalization of ADR tweet mentions to their corresponding meddra codes. For the first two steps, we reuse a state-of-the-art approach, focusing our contribution on the final entity-linking step. For that we propose a simple Few-Shot learning approach, based on pre-trained word embeddings and data from the UMLS, combined with the provided training data. Our system (relaxed F1: 0.337-0.345) outperforms the average (relaxed F1 0.2972) of the participants in this task, demonstrating the potential feasibility of few-shot learning in the context of medical text normalization.

C.1 Introduction

Team myTomorrows-TU Delft participated in subtask 3 of the 2019 Social Media Mining for Health Applications (SMM4H) [46] workshop, which is an end-to-end task. The goal is, given a tweet, to 1) automatically classify tweets containing an adverse drug reaction mention; 2) extract the exact ADR mention; 3) normalize the extracted ADR to its corresponding Medical Dictionary for Regulatory Activities (meddra) code. The task is evaluated based on strict and relaxed F-score, precision and recall.

From an NLP perspective, this task poses a significant challenge as there is a large gap between the informal language used in social media and the formal medical language.

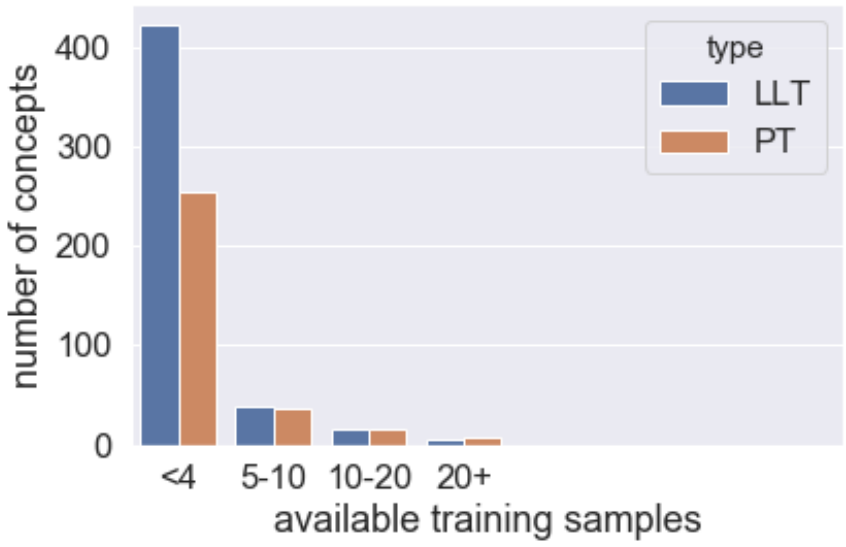


Figure C.1: Available training samples per the medical concept present in the training data

Moreover, there is an absence of large annotated datasets, and datasets which are available often suffer from class imbalance. Illustrating this, Figure C.1 provides an overview of the number of samples per class in the SMM4H task 3 dataset.

Our end-to-end system consists of existing state-of-the-art for the first two steps. We focus our efforts on the third -normalization- step, which we formulate as a Few-Shot Learning problem (FSL), following the definition by wang2019few [213]. In the following sections, we describe (1) the datasets that we worked on, (2) our approach in more detail and finally (3) our results and conclusions.

C.2 Data

Datasets

With the three subtasks, three manually annotated datasets were provided. All datasets contain tweets containing an ADR (positive) and without an ADR (negative). A brief overview of these datasets is provided in Table C.1, but for more context we refer to [46].

Preprocessing

The provided dataset for subtask 3 consists of ADR mentions, annotated with their corresponding MEDDRA code. In the hierarchy[1] of MEDDRA, one Preferred Term (PT) is

[1] https://www.meddra.org/how-to-use/basics/hierarchy

Task	Training data	
	#Positives	*#Negatives*
1	2374	23298
2	1212	1155
3	1212	1155

Table C.1: Statistics of the training data used for task 1, 2 and 3

linked to one or more Lower Level Terms (LLTs) which are more specific descriptions of the related concept.

The provided dataset contains a mix of PTs and LLTs, mapping the 1212 ADR mentions to more than 500 different codes. Observing that the evaluation of the workshop task is performed on PT level, we map all annotations to the corresponding PT, as a preprocessing step. After this preprocessing step, the 1212 training mentions are mapped to 319 MEDDRA codes. Figure C.1 provides an overview of the class distribution before and after preprocessing.

Prior Knowledge

In the training set for subtask 3, 149 out of the 319 MEDDRA codes that are present in the dataset (46.7%) have just one available training sample, while 254 (79.6%) have less than five training samples. To deal with the scarcity of samples, we create a prior knowledge dataset considering the 319 MEDDRA PTs in the training data. This dataset consists of the preferred names provided by the MEDDRA vocabulary and their corresponding preferred names in the Consumer Health Vocabulary (CHV), as mapped by the UMLS. The resulting dataset cointains 1,854 preferred names for the 319 MEDDRA codes.

C.3 Method

Our contributions focus on the normalization step, linking ADRs to their corresponding MEDDRA code. However, to be able to perform an end-to-end evaluation, we use existing state-of-the art techniques for subtask 1 [184] and 2 [35], which we train on the workshop datasets [2].

The state-of-the-art approach for medical concept normalization in user-generated text is deep-neural networks [121] which outperform traditional methods, when sufficient training data are available.

We trained both the CNN and RNN described by [121] on the dataset for task 3, finding that the RNN has the best performance. On closer observation (and not surprisingly), we found that the accuracy of the RNN drops when fewer samples are available in the training data, as depicted in figure C.2.

[2]For task 1, we trained using the suggested settings, assigning 3:1 class weight favouring the ADR class. For task 2, we trained using the pre-trained-fixed setting.

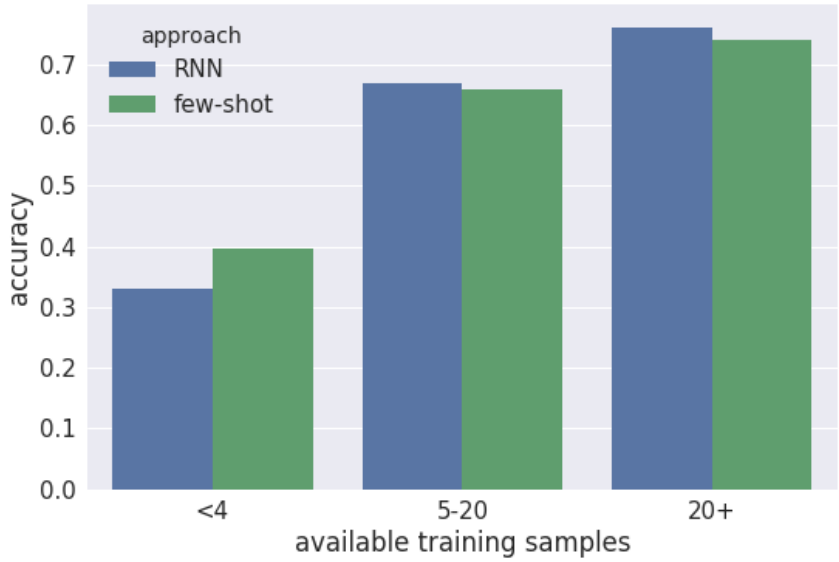


Figure C.2: Accuracy per number of training samples.

To deal with this drop in performance, we propose an embedding-based classifier that compares the ADR extracted mention to its 1-Nearest Neighbour on a vector space containing a) representations of the ADR mentions in the training data and b) representations of the prior knowledge dataset. Our intuition is that the embedding-based binary classifier would perform better on classes with a low number of samples, whereas an RNN would perform well on classes with higher sample numbers.

To create our embedding-based classifier we employ the pretrained Google News Word2Vec model [153]. Using this model, we create vector representations for the ADR mentions in our training data[3]. Similarly we create vector representations for the mentions gathered in our prior knowledge dataset. At test time, we employ the same Word2Vec model to create a vector representation of the unseen ADR mention. Using a 1-Nearest Neighbour (with cosine similarity as distance metric), we then select the corresponding MEDDRA concept. Figure C.2 shows that this model indeed seems less sensitive to low sample numbers.

For our experiments, we use 4 systems: (1) RNN: the RNN proposed by [121], trained on the both prior knowledge and the training set (which provides the best performance), (2) FSL: our 1-NN based on a combination of prior knowledge and the training set, (3) RNN+FSL (1): an ensemble of the RNN trained on only the training set and the FSL based on training + prior knowledge, and (4) RNN+FSL (2): an ensemble of the RNN trained on the training set and prior knowledge and the FSL based on training + prior knowledge. For our ensembles, we trust the model with the highest confidence (we used the cosine similarity for the 1-NN model to represent confidence) in case of disagreement.

[3]for mentions of more than one token we added the vectors

Technique	Relaxed			Strict		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
RNN	*0.318*	*0.337*	*0.327*	*0.232*	*0.246*	*0.239*
FSL	**0.336**	**0.355**	**0.345**	**0.237**	**0.252**	**0.244**
RNN+FSL (1)	*0.328*	*0.347*	*0.337*	*0.23*	*0.244*	*0.237*
RNN+FSL (2)	*0.331*	*0.35*	*0.34*	*0.235*	*0.249*	*0.242*
Task 3 AVG	*0.29*	*0.311*	*0.297*	*0.205*	*0.224*	*0.211*

Table C.2: Relaxed and strict Precision/Recall/F-score for RNN, FSL, RNN+FSL (1) and (2) and the average score of all the participated team in task 3 (Task 3 AVG)

C.4 Results

Our results are summarized in Table C.2. Despite the fact that the RNN+FSL performed better in our development set, it did not generalize in the test data. On the test and evaluation data, FSL outperformed all the other techniques and achieved a 0.345 relaxed F-score and a 0.244 strict F-score which are above the average performance achieved in this task by all participants (i.e. Task 3 AVG).

C.5 Conclusions

In this paper, we describe our approach in subtask 3 of the SMM4H shared task for normalization of Adverse drug reaction mentions in Twitter posts. Our few-shot learning approach performs above the average in this task and hence we believe it to be a promising approach in cases where the amount of training data is limited.

As future work, we will focus on the discrimination between the ADRs that belong to one of the 'commonly seen cases' (classes with sufficient training data) from the 'rare cases' (classes with insufficient training data). This will allow us to efficiently combine a deep neural network with a few-shot learning approach into a more robust system that successfully links ADR tweet mentions into its MEDDRA codes.

Appendix D

Facet Embeddings for Explorative Analytics in Digital Libraries

This appendix contains an example of the application of the extracted long-tail entities in the digital library domain.

Abstract

With the increasing amount of scientific publications in digital libraries, it is crucial to capture *"deep meta-data"* to facilitate more effective search and discovery, like search by topics, research methods, or data sets used in a publication. Such meta-data can also help to better understand and visualize the evolution of research topics or research venues over time. The automatic generation of meaningful deep meta-data from natural-language documents is challenged by the unstructured and often ambiguous nature of publications' content.

In this paper, we propose a domain-aware topic modeling technique called *Facet Embedding* which can generate such deep meta-data in an efficient way. We automatically extract a set of terms according to the *key facets* relevant to a specific domain (i.e. scientific objective, used data sets, methods, or software, obtained results), relying only on limited manual training. We then cluster and subsume similar facet terms according to their semantic similarity into facet topics. To showcase the effectiveness and performance of our approach, we present the results of a quantitative and qualitative analysis performed on ten different conference series in a Digital Library setting, focusing on the effectiveness for document search, but also for visualizing scientific trends.

D.1 Introduction

In light of the increasing amount of scientific publications, there is a growing need for methods that facilitate the exploration and analysis of a given research field in a digital library collection [133]. Existing approaches rely on word-frequency analysis [191], co-citation analysis [29], co-occurrence word analysis [94], and probabilistic methods like Latent Dirichlet Allocation (LDA) [75]. While popular, these approaches suffer from

one major shortcoming: by offering a generic solution, they fail to capture the intrinsic semantics of text related to a specific domain of knowledge. For instance, probabilistic methods like LDA are designed to be generic and widely applicable; however, they often miss out on topics that are relevant from a user's point of view.

To support richer retrieval experience, we advocate extracting *"deep meta-data"* from scientific publication, i.e. meta-data able to represent domain-specific properties and aspects (*facets*) in which a document can be considered and understood within its (research) domain.

Let us consider, for instance, the domain of *data processing and data science*, which is gaining popularity due to the availability of great amount of digital data, and progress in machine learning. In this domain, researchers and practitioners need to develop an understanding of the properties of available *datasets*; of existing data processing *methods* for the collection, enrichment and analysis of data; and of their respective implementations as *software* packages. The availability of deep meta-data about the facets (*datasets*, *methods*, and *software*) would enable rich queries like: *Which methods are commonly applied to a given dataset?*; *Discover state of the art methods for point of interest recommendation that have been applied to geo-located social media data with high accuracy results.* To the best of our knowledge, no state-of-the-art system is currently able to provide answers to the previous queries.

This paper presents an approach for generating domain-aware *"deep meta-data"* from collections of scientific publications. We focus on the data processing domain, and address the main facets described in the DMS ontology [140], namely *datasets*, *methods*, *software*, *objectives*, and *results*. We build upon a basic distant supervision approach for sentence classification and named entity extraction [144], and extend it with *facet embeddings* to automate the creation of *Facet Topics*, i.e. clusters of semantically similar facet terms which allow for easier querying and visualization. Our contributions are as follows:

- We introduce and formalize the concept of *facet topics*, which subsume a set of facet terms into higher level topics more suitable for exploration, visualization, and topic centered queries.
- We describe a novel approach for facet topic identification through *facet embeddings*. The approach combines distant supervision learning on rhetorical mentions for facet-specific sentence classification; semantic annotation and linking for facet terms extraction; and semantic clustering.
- We quantitatively and qualitatively assess the performance of our approach, and compare to established techniques like LDA topic modeling.
- We showcase our approach with a study exploring and visualizing trends and changes within the domain of data processing research, based on deep meta-data extracted from 11,589 research publications.

D.2 Related Work

The information overload in digital libraries is a crucial problem for researchers. Online digital libraries like the ACM Digital Library (DL), IEEE Xplore, CiteSeer etc, provide search options for finding relevant publications by using *"shallow"* meta-data such as the title, the authors, keywords or other simple statistical measures like the number of citations and download counts. However they are not designed to support the analysis of *"deep"* meta-data such as the topic, or methods and algorithms used in scientific publications.

There has been a large body of research focused on *deep* analysis of publications in scientific domains such as Software Engineering [133], Bio-informatics [197], Digital Library evaluation [4], or Computers science [88]; for different purposes, such as finding topic trends in a domain [133, 88] and evolution of scientific communities popularity [79]. Common approaches rely on methods such as word-frequency analysis [191], co-citation analysis [29, 88], co-word analysis [94], and probabilistic methods like latent Dirichlet allocation [75]. In contrast to existing literature which is either specially tailored to a domain or fully generic, our work combines the strength of both approaches by being partially domain-aware: after defining domain-aware facets using (limited) expert feedback, our approach automatically extracts topics by analyzing the co-occurrence of named entities related to the facets, thus is scalable within a domain while still taking advantage of domain-specific knowledge and peculiarities.

While most current research [133, 79, 191] limits the analysis of a publication's content to its title, abstract, references, and authors, we extract facet terms from the full text of scientific publications, in order to obtain more descriptive and accurate topics. In addition, our method is not only based on selecting the most frequent keywords (e.g. nouns, verbs set and proper nouns) [191], and, differently from probabilistic methods like Latent Dirichlet Allocation [75], it considers the semantics of terms for topic identification.

Some existing methods for domain-specific concept extraction and categorization are based on noun phrase chunking [79, 201] and use a bootstrapping approach to identify scientific concepts in publications. More recent research [193] used both corpus-level statistics and local syntactic patterns of scientific publications to identify and cluster similar concepts. Our method follows a distant supervision approach, a simple feature model (bags-of-words), and does not require prior knowledge about grammatical [201] and part-of-speech characteristics of facet terms. However, we do require a brief training phase for adapting our approach to a new domain.

D.3 Problem Description and Modeling

The goal of our work is to annotate n documents $D = \{d_1, ..., d_n\}$ of a domain-specific (scientific) corpus with faceted semantic meta-data. This meta-data goes alongside already available structured meta-data like for example author names, publication year, or citations. In particular, we aim at annotating documents with both *facet terms* and *facet topics*, as discussed in the following:

Facets and Facet Sets: The central elements of our approach are *facets*. Facets represent a perceived aspect relevant to user's understanding of documents in corpus D. When adapting our method to a given corpus, a *facet set* has to be defined which is used for describing documents in D, denoted as $F = \{f_1, f_2, \ldots, f_n\}$. Defining a good facet set requires some domain expertise. In the study presented in this work, we used specific facet set designed based on [140], namely the F_{DMS} facet set covering facets for a document corpus focused on data processing research. This facet set covers the five facets dataset, methods, software, objective, and result. We denote this as $F_{DMS} = \{DST, MET, SFT, OBJ, RES\}$.

Facets Terms: For each document $d \in D$ and facet $f \in F$, we extract a set of *facet terms* FT_f^d. A facet term $ft \in FT_f^d$ represents a term (usually a named entity, but also short phrases are possible) found in the full text of document d, and which can be clearly associated with facet f. We denote the set of all facet terms related to a given facet f found in any document of D as FT_f. Typical examples of facet terms for the method facet $MET \in F_{DMS}$ in our document collection are "Latent Dirichlet Allocation", "Support Vector Machine", or "Description Logic".

Facets Topics: Facet Terms are directly extracted from the full text of documents, and describe a document at a rather low level. In order to also allow for high-level analytics and queries, we introduce the concept of *facet topics*. Facet topics group multiple semantically related facet terms into a larger subsuming topic. In our use case scenario, when focusing on the methods facet, facet topics intuitively relate to research topics. For example, the terms "Support Vector Machine" and "Random Forest" can be subsumed by the facet topic "Machine Learning". The set of all facet topics for a given facet f is denoted as $FTP_f = \{t_1, t_2, \ldots t_k\}$, and each facet topic t is a subset of all facet terms, i.e. $t \in FTP_f : t \subseteq FT_f$. Furthermore, each term can be attributed to a topic, i.e. $FT_f = \bigcup_{t \in FTP_f} t$, and topics of a given facet are disjoint, i.e. $t_i, t_j \in FTP_f, t_i \neq t_j : t_i \cap t_j = \emptyset$ (however, there might be an overlap between topics of different facets, see next section). Terms in a facet topic show strong semantic cohesion.

D.4 Facet Term Extraction and Facet Topic Identification

In this section, we present our approach for *facet terms* and *facet topics* extraction from a collection of scientific publications, extending our previous work [144] by introducing additional steps for facet topic identification. An overview of our approach is shown in Figure E.3. Our approach is domain-aware in the sense that it requires some limited efforts to adjust it to a new domain (like deciding on facet sets), but is not inherently limited to a specific domain. In the following, we focus on the *data processing* domain, and address the five main *facets* (i.e. datasets, methods, software, results, and objectives) identified in the DMS ontology [140].

The process can base summarized as: First, we identify rhetorical mentions of a *facet* in the full text of documents. In this work, for the sake of simplicity, rhetorical mentions are identified at sentence level (i.e., each sentence is classified whether it con-

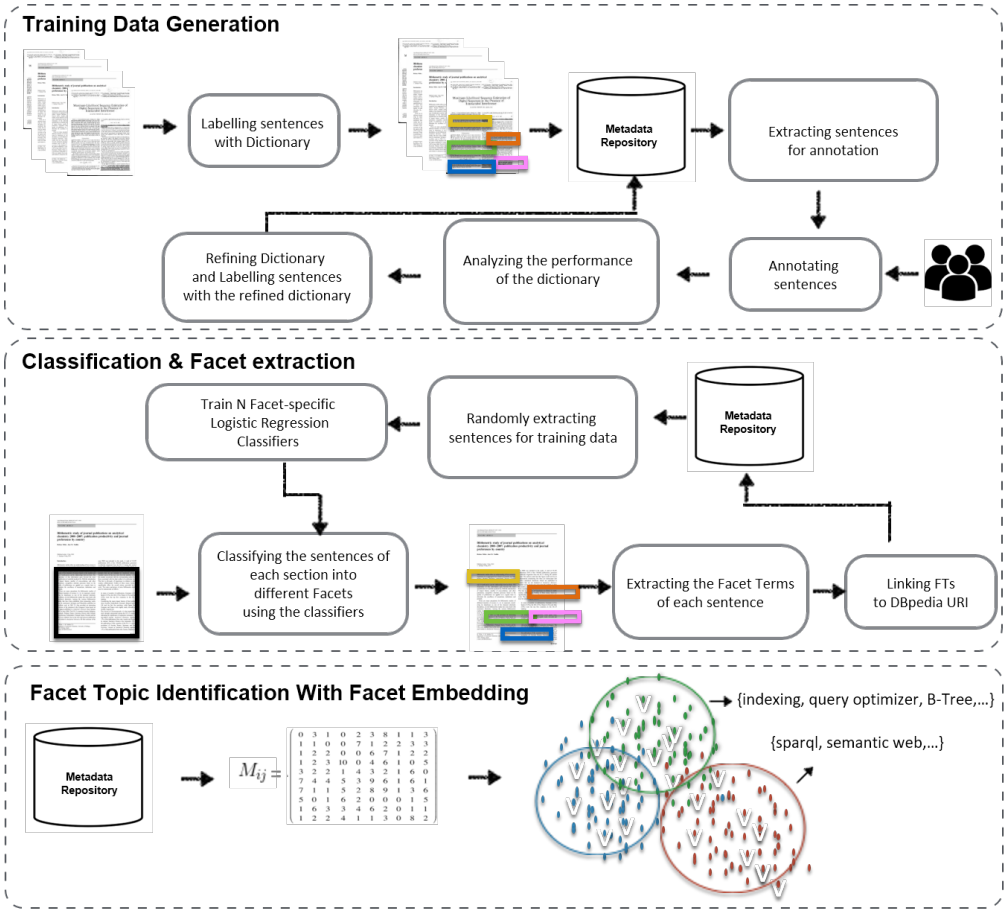


Figure D.1: Domain-aware Facet Modeling Workflow [144]

tains a rhetorical mention of a given facet or not). Future works will introduce dynamic boundaries, to capture the exact extent of a mention.

After a rhetorical mention was found, we extract potential *facet terms* from it. These terms are filtered and, when applicable, linked to pre-existing knowledge bases. Finally, all filtered facet term candidates finally form the document-specific facet term sets FT_f^d.

The identification of rhetorical mentions is obtained through a workflow inspired by distant supervision, a training methodology for machine learning algorithms that relies on very large, but noisy, training sets. The training sets are generated by means of a simpler classifier, for instance a mix of expert-provided dictionaries and rules, refined with manual annotations. Intuitively, the training noisiness can be canceled out by the huge size of the semi-manually generated training data. The method requires significantly less manual effort, while at the same time retaining the performance of supervised classifiers. Furthermore, this approach is more easily adapted to different application domains and changing language norms and conventions (more details in [144]).

Data Preparation: Scientific publications, typically available in PDF, are processed using state-of-art extraction engines, e.g. GeneRation Of BIbliographic Data

(GROBID) [129]. GROBID extracts a structured full-text representation as Text Encoding Initiative(TEI)-encoded documents, thus providing easy and reliable access paragraphs and sentences.

Test and Training Data Generation: We created training and benchmarking datasets for evaluating our rhetorical mention classifier by relying on a phrase dictionary for each facet (as described in [144]), automatically labeling all sentences in the document corpus if they contain a mention of relevant for a facet or not. Then, we randomly select a balanced set of 100 mentions of each facet. As the dictionary-based classifier is not fully reliable, we manually inspect and reclassify the selected sentences using feedback from two expert annotators, and rebalance the sentence set as needed. The inter-annotator agreement using the Cohen's kappa measure averaged over all classes was .58. Using this approach, we can create a reliable manually annotated and balanced test dataset quicker and cheaper compared to annotating whole publications or random sentences, as the pattern-classifier usually delivers good candidate sentences.

Machine-Learning-based Rhetorical Detection: As a next step in our distant supervision workflow, we train a simple binary Logistic regression classifier for each of the (facet) classes using simple TF-IDF features for each sentence. This simple implementation serves as a proof of concept of our overall approach, and can of course be replaced by more sophisticated features and classifiers.

As a test set, we use the aforementioned test set of 100 sentences for each facet. The *method* classifier showed the best performance with respect to its F-measure(0.71). From this, we conclude that our approach is indeed suitable for extracting *DMS* facet terms in a meaningful and descriptive fashion. However, there are still some false positives which cannot easily be recognized using simple statistic means, thus inviting further deeper semantic filtering in future works.

Facet Extraction, Linking, and Filtering: We extract *facet terms* from the labeled rhetorical mentions identified in the previous section, filtering out those terms which are most likely not referring to one of the *facet*, and retaining the others as an extracted term of the class matching the sentence label.

Facet extraction has been performed using the TextRazor API. TextRazor returns the detected *facet terms*, possibly decorated with links to the DBpedia or Freebase knowledge bases. As we get all *facet terms* of a sentence, the result list contains many *facet terms* which are not specifically related to any of the five *facets* (e.g. terms like "software", "database"). To filter the *facet terms*, we decided on a simple filtering heuristic assuming *facet terms* to be not relevant if they come from "common" English language (like software, database), while relevant terms are from domain-specific language or specific acronyms (e.g. SVM, GROBID). In our current prototype, we implement this heuristics by looking-up each term in *Wordnet*. Terms that can be looked-up are removed as we consider them common language. While this simple approach works for the "data science" domain, when extending our approach to a wider range of domains, this implementation should be replaced by more sophisticated heuristics, e.g., based on corpus statistics.

Facet Topic Identification With Facet Embedding After extracting all facet terms, we now strive to discover meaningful *facet topics*. Here, a central goal is to

subsume facet terms based on their semantic similarity. We implement a measurement for semantic similarity of terms by *Facet Embeddings*, which exploit co-occurrence of facet terms. For each $t_i, t_j \in FT_f$, we count how often these terms co-occur within the same document: $co_{t_i,t_j} = |\{d \in D : t_i \in FT_f^d \wedge t_j \in FT_f^d\}|$.

This results in a large (sparse) co-occurrence matrix. We reduce the dimensionality of the matrix using truncated Singular Value Decomposition. This step ensures the removal of less informative terms, while increasing the performance and usability of our approach (a smaller matrix is computationally cheaper to process). Using the reduced matrix, we now obtained an embedding of each facet term of a given facet (i.e., each term is represented as as row vector in the reduced co-occurrence matrix).

Finally, we now cluster all facet terms of a given facet in order to discover facet topics using K-means clustering, using Euclidean distance between the row vectors of two given terms as a distance measure. In order to find the optimal number k of clusters, we rely on Silhouette analysis, measuring the closeness of each point in a cluster to the points in its neighboring clusters. In addition to the Silhouette analysis, we also manually inspected the resulting clusters, but found that also from an qualitative point of view, the number of clusters determined by the Silhouette analysis is indeed the most satisfying one.

As a last processing step, we have two expert annotators label each facet topic in an iterative process until full agreement between the annotators was reached (see Section D.5 for more details).

We also implemented an alternative version of facet embeddings, relying on neuronal word embeddings (in our case word2vec [152]) instead of co-occurrence in rhetorical mentions. However, initial qualitative inspection of the results indicate that the distance measure between the term embeddings is an inferior representation of perceived similarity of facet terms from our experts' point of view. A deeper analysis of these results will be the subject of a later study.

D.5 Evaluation and Experimentation

In this section, we analyze the performance of our facet topic modeling workflow. We analyze and discuss the quality of facet terms extracted from the classified sentences. Next we qualitatively evaluate the quality of the topics extracted using Facet Embeddings. Finally we present some examples of information needs of researcher that can be fulfilled using our approach.

Corpus Analysis: We focused on 11,589 papers from ten conference series: The Joint conference on Digital Libraries (JCDL); the International Conference on Theory and Practice of Digital Libraries (TPDL); the International Conference on Research and Development in Information Retrieval (SIGIR); the Text Retrieval Conference (TREC); the European Conference on Research and Advanced Technology on Digital Libraries (ECDL); the International Conference on Software Engineering (ICSE); the Extended Semantic Web Conference (ESWC); the International Conference On Web and Social Media (ICWSM); the International Conference on Very Large Databases (VLDB); and the International World Wide Web Conference (WWW).

Conf.	Size		Rhetorical sentences						Unique Facet Terms				
	Years	#PUB	#SNT	#OBJ	#DST	#MET	#SFT	#RES	#OBJ	#DST	#MET	#SFT	#RES
ESWC	2005-2016	626	84439	12725	13528	26337	9614	22245	4197	4910	6987	4557	6416
ICWSM	2007-2016	810	34987	6096	4277	8936	1830	13848	2830	2241	3658	1538	4499
VLDB	1975-2007	1884	272380	30360	56647	77123	13317	94933	8008	13207	15319	6262	17532
WWW	2001-2016	2067	322801	47134	40449	97760	21347	116111	10902	10917	17783	8863	19822
ECDL	1997-2010	820	65470	12008	8079	18638	8130	18615	4634	3650	5894	4125	5376
ICSE	1976-2016	1834	182029	29850	16284	57494	26042	52359	8169	5841	12503	8776	11728
JCDL	2001-2016	1416	99747	19290	13002	27786	9692	29977	6524	5240	7754	5037	7979
SIGIR	1971-2016	412	39688	5080	4813	13214	2050	14531	2144	2377	4126	1588	4068
TPDL	2011-2016	276	23176	4660	3342	6032	2489	6653	2168	1871	2625	1719	2503
TREC	1999-2015	1444	122456	11828	14760	39121	8825	47922	6616	3085	4095	3286	7668

Table D.1: Quantitative analysis of the rhetorical sentences and facet terms extracted from ten conference series. Legend: PUB (publications), SNT (sentences), OBJ (objective), DST (dataset), MET (method), SFT (software), RES (results)

Due to changes in publication platforms and PDF format, the corpus does not contain all publications of each conference.[1] We believe the absence of few publications not to have an impact on the significance of our findings, but might still be reflected in the shown diagrams and results. Table D.1 provides basic statistics for the analyzed corpus, including: the range of years, the number of publications, the number of extracted rhetorical sentences and mentions, and the distinct facet terms extracted from rhetorical sentences. *Method* and *results* facets are the most frequent, followed by *objectives*.

Quality of extracted topics: We investigated or domain-aware facet embedding compared to the domain-independent technique Latent Dirichlet Allocation (LDA) by asking two domain experts to label the topics derived by each method, while assessing which are more meaningful. For the sake of presentation, we set the maximum number of topics to $T = 30$, and performed the Silhouette analysis to find the number of optimal topics, which resulted in 27 topics.

In order to allow for a more informative comparison, we applied both approaches to the full text of publications, and also to only pre-classified sentences (because LDA is usually applied to full texts. Thus, in one case we use our facet embedding without restricting to classified facet sentences, but we also consider a case where LDA is applied to the set of all sentences which belong to a given facet). For the sake of brevity, we consider only the *method* facet when performing a facet pre-classification of sentences. The *method* classifier has shown the best performance with respect to its F-measure. Our analysis shows comparable results with the other facets.

Full Text without Facet Classification: For full text experiments, the corpus has been pre-processed by removing stop words, and representing each document as a bag-of-words. We use the LDA implementation provided by the `scikit-learn` library. For compatibility, we trained for 27 topics. For evaluating facet embeddings without any domain specific pre-classification on full texts, we are assuming that there is only a single facet, and each sentence of a document is classified as such (note: this is not how we usually intend our method to work).

Consider only Sentences classified as Method facet: In this experiment, we perform

[1]For instance, around 100 JCDL papers for 2014 are not included in the analysis, as the proceedings were, only for that year, published by `ieee.org`

Full text	*LDA*	reference, abstracts, linking, sofm, similarity annotations, backup, linkservice, annotation, digital query, data, user, web, information
	FE	sparql, semanticweb, linkeddata, rdf, dbpedia, sql,relationaldatabase, tuple, queryoptimization, datawarehouse, socialnetwork, facebook, randomwalk, pagerank, powerlaw
Facet	*LDA*	documents, used, classification, libraries, digital measure, performance, given, recommendation, used, social, twitter, media, popular, past
	FE	searchalgorithm, timecomplexity, datastructure, dynamicprogramming, sparql, semanticweb, linkeddata, dbpedia,rdfs, socialmedia, lda, latentdirichletallocation, topicmodel, socialnetwork

Table D.2: Example top terms extracted using the generic (LDA) and domain-aware (FE) topics, using either full texts or only those sentences related to the *method* facet

Topic Name	***Top five terms***
Social Media Analytics: Text-based	social media, lda, latent dirichlet allocation, topic model, social network
Semantic Web: Knowledge Engineering & Representation	sparql, semantic web,linked data, dbpedia, rdfs
Semantic Web: Logic	description logic, dl, abox, tbox, semanticweb
Misc Topics: Web Information Systems	information retrieval, data structure, dataset, natural language, electronic media
Databases: Query Processing	tuple, hash join, sort, relational database, hash table
Databases: Modelling	data model, sql, query language, query optimization, tuple
Web Technologies	side, client, server, javascript, web application
Digital Libraries	digital library, information retrieval, xml,user interface, computer science
Machine Learning	machine learning, support vector machine, supervised learning, dataset, information retrieval
Web Engineering: P2P & Distributed Systems	peer, to, ip address , rdf, webservice
Social Graph Algorithms	greedy algorithm, approximation algorithm, optimization problem, social network, electronic media
Social Graph Analysis	pagerank, random walk, social network, webpage, adjacency matrix
XML Databases	xml, xpath, xquery, xmlschema, sql
Software Engineering: Testing & Formal Methods	source code, test case, control flow, test suite, program analysis
Software Engineering: Systems	software development, software engineering, software development process,software system, case study
Web Engineering: System Modelling	use case, web service, model checking, case study, semantic web
Web Engineering: Client-Side	web page, user interface, web browser, web content, javascript
Information Retrieval: QA, NLP, and Complex Queries	trec, question answering, document retrieval, information retrieval, query expansion
Information Retrieval: Evaluation	adhoc, trec, query expansion, information retrieval, relevance feedback
Information Retrieval: Ranking	query expansion, language model, relevance feedback, trec, information retrieval
Information Retrieval: Mining	score, f1, supervised learning, crf, bic
Microsoft Technology	microsoft, microsoft sqlserver, sql, xml, microsoft word
Databases: Indexing	tree, trees, data structure, access method, search algorithm
Databases: Transaction Mangement	concurrency control, lru, serializability, aries, tion
Databases: Algorithms	search algorithm, time complexity, data structure, dynamic programming, dataset
Recommendation	collaborative filtering, recommender system, gradient descent, singular value decomposition, social network
System Engineering: Architecture	operating system, programming language, file system, data structure, software engineering

Table D.3: Top five *method* terms for each facet topic. Topic labels have been assigned manually by two experts.

the *facet topic* extraction as described in section D.4, including facet-based sentence classification, facet term extraction, and facet embedding, but limited to only the *Method* facet. As a comparison, we also perform LDA on only those sentences classified as methods (therefore also giving LDA the chance to take advantage of the domain-aware training).

Results: A manual inspection on the resulting topics show that those identified by LDA are very hard to label and are perceived as semantically less meaningful by our experts, while the topics based on Facet Embeddings produced coherent and interpretable topics which were perceived as understandable and useful. In table D.2, we provide an example of 3 randomly selected topics for each aforementioned experimental setup. It can be observed that topics generated from sentences pre-classified as *method* show better semantic cohesion than those generated from full texts. Furthermore, we provide the full result of labeling all 27 topics for the method facet in Table D.3. The top-40 term can be found in the companion website[2]

Application Example: Scientific Publication Retrieval: In this section we show scenarios of how computer science researchers could use our approach for their work. Furthermore, all faceted deep meta-data used in those scenarios has been published as

[2]http://www.wis.ewi.tudelft.nl/tpdl2017

Paper title	*Dataset and Method facet*
Personalized Interactive Faceted Search [109]	IMDB(DST), Faceted search(MET)
refeREE: An Open Framework for Practical Testing of Recommender Systems using ResearchIndex [40]	IMDB(DST), Recommender system(MET)
The Party is Over Here: Structure and Content in the 2010 Election [125]	Facebook(DST), Sentiment analysis(MET)

Table D.4: Examples of papers applying methods (MET) to given datasets(DTS)

an RDF knowledge base according to the DMS ontology, accessible from a SPARQL endpoint on the companion website.

Find publications that applied method X on a given dataset: Table D.4 shows the result of an example query for methods which have been applied to movie dataset (i.e. IMDB and MovieLens) or Social media data (i.e. Facebook). For instance, [40] is a paper containing both the facet terms "Recommender system" labeled as *method*, and "IMDB" labeled as *dataset*.

Retrieve the most used methods of a given conference series: To answer this question, we use the number of papers for each *method* facet topic shown in Table D.3 for a given conference. Results are shown in Figure D.2. The value in each cell denotes the values normalized by the number of publications in each conference overall. The figure also demonstrate the quality of our approach: topics like "Machine Learning" and "Information Systems" are popular for all considered conferences. "Database" topics are mostly popular in the VLDB conference series, while the topic "Digital Library" appears in JCDL and TPDL. Clearly, the extracted facet topics match the research focus of each conference. Also, other popular topics can be explored: conferences like JCDL or TPDL favor methods like Machine Learning, Digital Libraries, Web Information Systems, and Information Retrieval.

What are the trends for methods?: In order to answer this question, we visualize the number of publications covering a *method* facet topic (as listed in Table D.3) over the course of the last 10 years. The results are shown in Figure D.3, giving an intuition about the quality of our approach: e.g., methods related to machine learning, software testing, or social media analytics gained great popularity in the last 10 years; while, as expected, topics related to core databases techniques or XML processing are becoming less popular.

D.6 Summary and Outlook

This paper presents the design and evaluation of a novel method for domain-aware topic identification for collections of scientific publications. Our method aims at improving the ability of digital libraries systems to support the retrieval, exploration, and visualization of documents based on topics of interest. In contrast to previous work, is taking advantage of some domain-specific insights which vastly improves the quality of the resulting topics, while still being adoptable to other domains by minimal efforts.

Our proposed method relies on a combination of sentence classification, semantic annotation, and semantic clustering to identify *Facet Topics*, i.e. clusters of semantically related *terms* that are tied to an *facet* relevant to an user's understanding of a document

	ECDL	JCDL	TPDL	ICSE	VLDB	SIGIR	TREC	ICWSM	WWW	ESWC
Databases: Algorithms	0.0214	0.0249	0.0217	0.0176	0.1075	0.0362	0.0161	0.0288	0.0369	0.0264
Databases: Indexing	0.0045	0.0031	0.0007	0.0031	0.0758	0.0079	0.0015	0.0057	0.0069	0.0033
Databases: Modelling	0.0291	0.0116	0.0245	0.0292	0.1081	0.0225	0.0083	0.0076	0.0159	0.0279
Databases: Query Processing	0.0086	0.0056	0.0035	0.0117	0.1465	0.0150	0.0054	0.0057	0.0116	0.0160
Databases: Transaction Management	0.0024	0.0020	0.0035	0.0073	0.0728	0.0049	0.0012	0.0014	0.0043	0.0038
Digital Libraries	0.1909	0.1577	0.1058	0.0089	0.0063	0.0154	0.0076	0.0104	0.0116	0.0122
Information Retrieval: Evaluation	0.0122	0.0049	0.0014	0.0093	0.0046	0.0300	0.0797	0.0042	0.0061	0.0083
Information Retrieval: Mining	0.0003	0.0043	0.0014	0.0004	0.0016	0.0000	0.0017	0.0057	0.0052	0.0026
Information Retrieval: QA and NLP and and Complex Queries	0.0410	0.0282	0.0182	0.0085	0.0096	0.0780	0.2392	0.0212	0.0164	0.0191
Information Retrieval: Ranking	0.0395	0.0439	0.0357	0.0093	0.0098	0.1054	0.2548	0.0434	0.0299	0.0217
Machine Learning	0.0689	0.1415	0.1458	0.0542	0.0355	0.1592	0.1131	0.1619	0.1296	0.1070
Microsoft Technology	0.0217	0.0137	0.0070	0.0203	0.0160	0.0026	0.0073	0.0019	0.0143	0.0086
Misc Topics: Web Information Systems	0.2591	0.2549	0.2614	0.2067	0.1806	0.2513	0.1304	0.2011	0.1790	0.1215
Recommendation	0.0125	0.0280	0.0154	0.0080	0.0058	0.0450	0.0117	0.0477	0.0461	0.0191
Semantic Web: Knowledge Engineering & Representation	0.0178	0.0228	0.0596	0.0071	0.0114	0.0150	0.0100	0.0151	0.0449	0.2395
Semantic Web: Logic	0.0068	0.0011	0.0028	0.0026	0.0021	0.0009	0.0010	0.0024	0.0144	0.0753
Social Graph Algorithms	0.0042	0.0038	0.0077	0.0050	0.0152	0.0141	0.0041	0.0198	0.0417	0.0060
Social Graph Analysis	0.0255	0.0390	0.0294	0.0089	0.0154	0.0454	0.0327	0.0760	0.0664	0.0298
Social Media Analytics: Text-based	0.0065	0.0296	0.0308	0.0083	0.0052	0.0454	0.0151	0.2587	0.0561	0.0205
Software Engineering: Systems	0.0237	0.0181	0.0182	0.1705	0.0131	0.0123	0.0078	0.0109	0.0131	0.0176
Software Engineering: Testing & Formal Methods	0.0098	0.0060	0.0070	0.2095	0.0107	0.0093	0.0066	0.0071	0.0196	0.0136
System Engineering: Architecture	0.0175	0.0078	0.0112	0.0397	0.0216	0.0097	0.0034	0.0024	0.0097	0.0036
Web Engineering: Client-Side	0.0748	0.0658	0.0736	0.0283	0.0198	0.0498	0.0202	0.0387	0.0821	0.0350
Web Engineering: P2P & Distributed Systems	0.0116	0.0069	0.0021	0.0024	0.0099	0.0062	0.0034	0.0019	0.0134	0.0100
Web Engineering: System Modelling	0.0359	0.0307	0.0736	0.0892	0.0194	0.0093	0.0071	0.0109	0.0486	0.1063
Web Technologies	0.0062	0.0125	0.0112	0.0207	0.0048	0.0022	0.0027	0.0033	0.0389	0.0102
XML Databases	0.0478	0.0314	0.0266	0.0131	0.0707	0.0071	0.0080	0.0061	0.0371	0.0353

Figure D.2: Heatmap showing the relation between *research methods* and conferences. The values are normalized based on the numbers of papers in each conference.

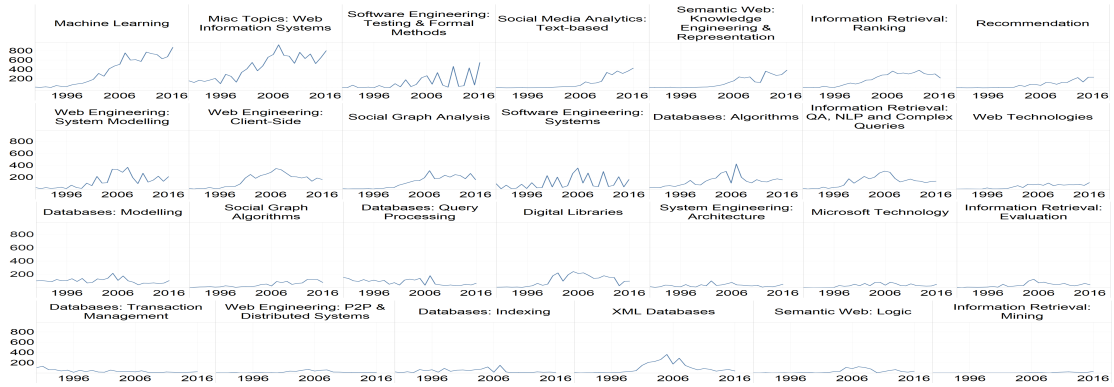


Figure D.3: The trends of *research methods* over years. The y axis shows the contribution of each topic in a certain year by means of the number of method-occurrence

collection. The method specializes on the extraction of facet-specific information through the classification of rhetorical mentions in sentences. A lightweight distant supervision approach with low training costs (compared to traditional supervised learning) and acceptable performance, allows for simple adaptation to different domains. Facet terms are extracted from candidate sentences using state-of-the-art semantic annotation tools, and are filtered according to their informativeness. *Facet Topics* are identified using a novel *Facet Embedding* technique.

We applied this novel method to a corpus of 11,589 publications on *data processing* from 10 conference series, and extracted metadata related to the 5 facets of the DMS [140] ontology for data processing pipelines. An extensive set of quantitative and qualitative

analysis shows that, despite its simple design, our methods allows for topic identification performance superior to state-of-the-art topic modeling methods like LDA.

While promising, results leave ample space for future improvements. We are interested in investigating the performance of more complex machine learning classifiers (e.g. based on word-embeddings), possibly applied to group of related sentences. We also plan to investigate new techniques for facet terms extractions, and study the performance of our approach with larger amount of *Facet Topics*. Finally, we plan to expand our analysis to additional domains, and investigate new facets of interest.

Appendix E

Conceptual Modelling: DMS Ontology

This appendix contains an ontology to support the description and encoding of relevant properties of long-tail entities found in scientific publications.

Abstract

The rise of Big Data analytics has been a disruptive game changer for many application domains, allowing the integration into domain-specific applications and systems of insights and knowledge extracted from external big data sets. The effective "injection" of external Big Data demands an understanding of the properties of available data sets, and expertise on the available and most suitable methods for data collection, enrichment and analysis. A prominent knowledge source is scientific literature, where data processing pipelines are described, discussed, and evaluated. Such knowledge is however not readily accessible, due to its distributed and unstructured nature. In this paper, we propose a novel ontology aimed at modeling properties of data processing pipelines, and their related artifacts, as described in scientific publications. The ontology is the result of a requirement analysis that involved experts from both academia and industry. We showcase the effectiveness of our ontology by manually applying it to a collection of publications describing data processing methods.

E.1 Introduction

Big Data analytics contributed to improvements in the state-of-the-art of several domains. Domain-specific data processing workflows (or "pipelines") facilitate the integration (at scale) of rich and meaningful knowledge mined from third-party, domain-agnostic data sources, thus often opening the field for before unseen innovation.

In this respect, social media data represents a common yet successful example; the collection and analysis of users' activities enabled novel studies in: urban planning (e.g. activity spaces analysis through points-of-interest mining [96]); public health care (e.g.

real-time monitoring of diseases diffusion [27]); marketing (e.g. analysis of consumer-brand relationships [91]); or pharmacovigilance (e.g. discovery of adverse effects of drugs [182]).

We refer to this powerful practice of integrating external data sources, by means of processing pipelines, and to extend and supplement the power of an information system for achieving new goals, as "Big Data injection". Efficient and effective injection of external data sources is not a straightforward activity: to build novel solutions, practitioners and data scientists are required to have a deep understanding of the properties and limitations of the available data sources; of existing data processing pipelines devoted to the collection, enrichment and analysis of data; and of their respective implementations. We argue that the lack of suitable models and tools able to encode and collect such knowledge is one of the main roadblocks for a more principled and widespread adoption of external data injection.

In this work, we focus on (scientific) publications as a primary source of knowledge related to data sources and data processing pipelines. Example of knowledge commonly contained in publications include: 1) the properties of data sets of interest (e.g. size, sparseness, diversity, or bias); 2) the properties and limitations of related data processing techniques (e.g. complexity, accuracy); and 3) the properties of software and tools for data processing (e.g. run-time performance). Unfortunately, this rich knowledge is not readily accessible, as it is distributed across a vast repository of unstructured natural-language documents. To the best of our knowledge, no state-of-the-art computer system is currently able to provide an answer to the following query: *"Find the methods for POI (Point-Of-Interest) recommendation on 4Square data having a precision no lower than 10% from the state of the art"*. A first step towards the creation of a system able to answer such query is the availability of a knowledge representation models (e.g. an ontology) able to capture relevant properties of data sources, methods, and software that are relevant for Big Data Injection purposes.

Previous work tackled in several ways the representation, with ontologies, of some aspects of data sources and data processing pipelines in scientific publications [19][100]. However, to the best of our knowledge, no existing ontology is able to capture all the classes, properties, and relationships needed in order to answer the aforementioned query.

In this work, we propose a novel ontology called `DMS` [1] (Dataset, Method, and Software) able to encode and describe properties of data processing pipelines for external data injection in a machine-readable way. We elaborate on the requirement elicitation process that lead to the creation of the `DMS` ontology; the process included an expert study, and an extended analysis of the state-of-the-art of related ontologies. We showcase the effectiveness of our ontology by manually annotating a collection of publications describing data processing methods, showing that indeed all relevant information can be captured. Finally, we outline and discuss our vision in how this ontology will be integrated into a larger ecosystem including sophisticated information extraction and reasoning. The ontology could serve multiple use cases. In this paper we focus on semantically rich queries. However, we also envision scenarios where practitioners could

[1] Supporting website: `http://www.wis.ewi.tudelft.nl/DMS_SWM2017`

be supported with designing Big Data injection workflows by means of analytics on repositories of publications and data processing pipelines.

E.2 Requirement Elicitation

In this section, we will elaborate on the activities that led to the elicitation of the requirements that concluded in the current version of the DMS ontology.

Methodology

The design of the DMS ontology for data processing pipelines description has been performed according to the *Methondology* guidelines presented in [63].

To scope the requirement elicitation activity, we identified a domain of interest relevant for external data injection, namely "social media data analysis". As also argued in the introduction, this domain finds widespread application and attracted considerable academic and industrial interest. In addition, data processing pipelines for social media data feature a full range of activities: from data set creation (e.g. crawling) and analysis, to the design of novel data enrichment methods (e.g. semantics of locations); to the adoption of existing methods and software (e.g. LIWC, Twitter API).

To capture the perspectives of both producers and consumers of publications related to data sources and pipelines, the requirement elicitation process involved two classes of relevant actors: *Data Science Practitioners*, and *Data Scientists* from academia.

We engaged with practitioners to discuss and identify relevant use cases for Big Data Injection, in the domain of social media data analytics. Scientists were interviewed in order to collect knowledge about the information that could be found in scientific publications, and that could be considered relevant for external data injection purposes from a scientific point of view.

Identification of Industrial Case Studies

We interacted with practitioners from the *Data Science & Analytics* unit of *Capgemini Netherlands*. Being involved with tasks related to data and data processing, the unit is a relevant and informed party for investigation. After an initial brainstorming on relevant use cases, we focused on the *Searching* use case, i.e. the task of retrieving, from collections of scientific publications, relevant information about available data sources and data processing pipelines.

By means of semi-structured interviews, we derived a set of information needs (queries) related to the discovery of knowledge about data set, method and software from (scientific) publications. Examples of derived information needs include:

Searching for data sets: Researchers and practitioners are often looking for innovative applications of known data sets to new applications. Here, a typical query would be: *Find the available Web data sets that contain demographic information, but that have never been used in our organisation to study cultural differences across Dutch cities.*

Searching for methods: Researchers and practitioners often have to decide which methods will satisfy the domain requirements with respect to pre-defined metrics, e.g. precision and reliability. A typical sample query would be: *Find the method for POI recommendation based on Matrix Factorisation that features the best AUC metric in literature.*

Searching for software: On a similar note, researchers and practitioners are often interested in comparing software implementing aforementioned methods with respect to properties like performance or scalability. A typical example query would be: *Find the software used to tag objects in the images of the social media data (e.g. Instagram) with a precision within 20% from the state of art but with image annotation time lower than 100 milliseconds*

These examples of information needs clearly hint to three core functional requirements: 1) the ability to extract from a source of knowledge (e.g. a publication), preferably in an automated way, the information nuggets that contain relevant information about data sets and data processing pipelines; 2) the ability to link such information nuggets across different resources (e.g. publications, public or legacy databases); and 3) the ability to reason upon a body of knowledge, so as to infer properties that are not directly encoded in the original resource (e.g. the property of being "state-of-the-art").

Expert Analysis of Scientific Publications

For a system to satisfy the requirements 2) and 3) described above, the information nuggets contained in a publication must be first identified. Their identification allows the distillation of a set of concepts, properties, and relationships, that will constitute the main elements of the structured representation of the information contained in publications.

To this end, the authors interviewed two data scientists operating in the field of database systems and information retrieval. The two experts operate in our faculty, and were selected based on their academic and industrial experience with data processing pipelines. We selected five relevant and recent publications [47, 90, 112, 164, 214], and we asked the two experts to annotate them. This selection focuses on papers with a complete coverage of the respective data processing pipeline and its context in the domain of interest.

The annotation tasks required the highlighting – with different colours – of paragraphs (or sentences) containing information relevant information about data sets, data processing pipelines, and the methods and software therein developed or employed. The scientists were also required to complement the annotation with a free-text description of the relevant attributes contained in the text (e.g. size of data set, parameter of used methods, link to software).

We manually processed the experts annotations. We observed overlaps between highlighted paragraphs, as well as some differences in terms of the level of details in free-text annotations. We interpreted overlapping highlights as a signal of relevance for the annotated text. Also, we extracted relevant terms from the free-text annotations, and resolved synonymity among terms.

E.3 Ontology design

The requirements elicitation process led to the identification of four core concepts relevant for describing knowledge about big data data sources and processing pipelines in *publications*: *data sets*, *methods*, *software implementations*, and *experiments*.

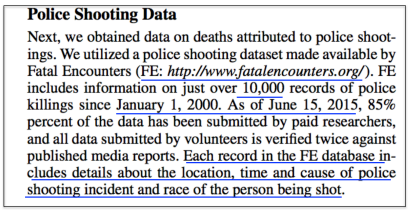
While the first three concepts are evidently relevant, the fourth deserves clarification. When considering the example queries provided by the industry practitioners in Section E.2, it clearly emerges the strict relationship that exists between several aspects of data processing (e.g. pipelines) and the experimental set-up described in a research paper. That is, in order to realise a certain research `objective`, an `experiment` is instrumented where a specific `combination` of methods is applied to a data set as part of a data processing pipeline, thus achieving a specific `performance` and `result` in that `context`.

In this section, we describe the final conceptualization of the `DMS` ontology, based on a term-extraction process (Section E.3) and also by studying and integrating existing ontologies (see section E.3) that are related, but not sufficiently expressive to cover the needed concepts, properties and relationships. Section E.3 describes the resulting `DMS` ontology.

Term extraction

Term extraction is a central step in ontology engineering, in which the key concepts of the ontology and their characteristics are identified. We base our term extraction on the expert interviews, and their annotations as discussed in the previous section.

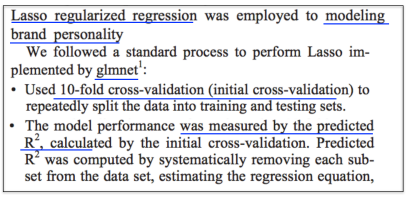
Figure E.1 depicts an excerpt from [47], showing rhetorical phrases annotated by one of the experts. These phrases encode characteristics of the data set used by the publication the excerpt was taken from (highlighted in blue), for instance: where to obtain the dataset, its size, and its temporal coverage. The goal of term extraction is to identify ontology terms and concepts which can explicitly encode the desired information in such phrases, for a large library of documents.



Police Shooting Data

Next, we obtained data on deaths attributed to police shootings. We utilized a police shooting dataset made available by Fatal Encounters (FE: *http://www.fatalencounters.org/*). FE includes information on just over 10,000 records of police killings since January 1, 2000. As of June 15, 2015, 85% percent of the data has been submitted by paid researchers, and all data submitted by volunteers is verified twice against published media reports. Each record in the FE database includes details about the location, time and cause of police shooting incident and race of the person being shot.

Figure E.1: Text excerpt with mentions of data source attributes highlighted.



Lasso regularized regression was employed to modeling brand personality

We followed a standard process to perform Lasso implemented by glmnet[1]:

- Used 10-fold cross-validation (initial cross-validation) to repeatedly split the data into training and testing sets.
- The model performance was measured by the predicted R^2, calculated by the initial cross-validation. Predicted R^2 was computed by systematically removing each subset from the data set, estimating the regression equation,

Figure E.2: Example text containing information about a method and a software.

In the following, we focus on the *properties* of the core concepts covered by `DMS`. We started by collecting all raw terms related to properties of the concepts mentioned by the experts during the interview. For the annotations (as in Figure E.1), we assigned a label from an uncontrolled vocabulary to all highlighted rhetorical phrases which best describes the encoded property. As a next step, we then manually grouped all resulting

terms and labels with respect to their semantics, and finally subsumed each group into one property as shown below.

During this process, we identified some additional concepts that are important to describe how data sets, methods, and software interact as parts of a Big Data injection workflow (as for example, some method is *applied* to a data set in an *experiment* which has a very specific *objective*.) We discuss these meta- and auxiliary concepts in section E.3.

px

Data sets used or created in a publication. Data sets can be described by means of:

- The schema properties of the data set, such as the set of attributes (i.e actual data stored into a file, like a JSON file which contains Twitter data with date, time, user, and content.)
- Quantitative properties of the data set, such as the size, and descriptive statistics like sparsity or skew.
  In our annotated publications, these properties are often encoded in tables.
- Temporal and Spatial properties of the data set (often found in text, e.g. "data gathered between October 2009 and September 2011 from the French region in Switzerland [159]").
- The application of the data set (also usually found in text, e.g. "tracking Twitter for public health").
- The scope of the data set (e.g. social media data, census data).
- The URL linking to the location of the data set (this is often found in text, footnotes, or references).
- The license (e.g. "public domain")

px

Methods, i.e. algorithms (novel or pre-existing) used to create, enrich, or analyse a data set. Methods can be described by means of:

- The parameters (often found in text, e.g. "we used 10 fold cross-validation").
- The data sets and parameters used or created by the method.
- Reference to an existing method (e.g. reference to another paper, references to implementing software)
- The application of the method (e.g. "Lasso regularized regression was employed to modeling brand personality")
- The result of the employed method (e.g."the model predicted R^2 values as high as 0.67", which is also usually found in form of tables or figures).

px

Software, i.e. computer tools employed to support the creation or the processing of data. Software can be can be described by means of:

- The result produced by the software.
- The license (e.g. "public domain").
- The application of the software (e.g. "emotional expression measures were computed using LIWC").
- The URL linking to the download location of the software.
- The programming language used.
- The performance of the software in the context of the experiments.

Our findings are summarised in table E.1. The requirements elicitation activity highlights the need for the representation of data set properties, along with provenance information with respect to their creation and processing, and their relationship with methods and software organised in data processing pipelines designed for specific usage contexts.

	DMS [2]	**DOCO**[39]	**DEO**[3]	**Disco** [19]	**CiTo** [170]	**OntoSoft** [67]
Describing Data sets						
Variables-Data files	+	-	-	+	-	-
Quantitative properties	+	-	-	+	-	-
Temporal and Spatial	+		-	+	-	-
Scope	+	-	-	+	-	-
License	+	-	-	+	-	-
Link to location	+	-	-	-	-	-
Describing methods						
Methods	+	-	+	-	-	-
Method parameters	+	-	-	-	-	-
Results	+	-	+	-	-	-
Application	+	-	-	-	-	-
Citation	+	-	-	-	+	-
Describing software						
Programming language	+	-	-	-	-	+
Average runtime	+	-	-	-	-	+
Describing Experiments						
Objective	+	-	-	-	-	-
Research Questions	+	-	-	-	-	-
Figures-table	+	+	-	-	-	-

Table E.1: Comparison established ontologies with respect to our three core-topics, and publication specific meta properties. Plus sign (+): The property has been covered by the ontology, Minus sign (-): The property has not been covered by the ontology. ([2]Our proposed ontology, [3]`http://purl.org/spar/deo`) .

Reuse of existing ontologies

One design goal of DMS is to rely on the lessons-learned of established ontologies, and reuse their vocabulary whenever possible. Therefore, we provide an overview of the current state-of-the-art of related ontologies with a focus on the previously identified requirements in Table E.1, and discuss them with respect to the three core concepts of publications, methods, data sets, and software in the following.

Describing scientific publications and methods. In this work, we focus on properties of data sources and processing pipelines as extracted from scientific publications. Many aspects of the nature of the overall data processing pipeline are described in the rhetoric of the research publication itself (like in the motivation, abstract, or discussion). Several ontologies exist for describing structural properties (e.g title, sections, header, etc.) and rhetorical elements (e.g contribution, results, figures, tables and etc) of scientific publications. The Semantic Publishing and Referencing Ontologies (SPAR Ontologies)[4], is one of the first attempts to describe different aspects of semantic publishing and referencing in a machine-readable format. SPAR consists of 13 OWL2 DL ontology modules. For the sake of brevity, we only describe the ones (Doco, Deo and Cito) that are related to our properties of interest. The Document Components Ontology (Doco) [39] provides a structured vocabulary for both structural (e.g. block, inline, paragraph, section, chapter) and rhetorical (e.g. introduction, discussion, table, reference list, figure, appendix) components of the paper. Doco imports Discourse Elements Ontology (Deo)[5] which provides a vocabulary for rhetorical elements within documents, including *methods* and results. DoCo and Deo both complement each other. Ruiz-Iniesta in [180] reviewed the scholarly document ontologies and suggested Doco and Deo for describing the structural and rhetorical elements of the publications, and the Citation Typing Ontology (Cito) [170] for describing the citation acts between the scientific publications (e.g. cito:cites, cito:extends, cito:isDescribedBy). Notice that the mentioned ontologies do not directly address properties of data sets or software.

Ontologies for describing data sets. The DDI-RDF Discovery Vocabulary (Disco) [19] and RDF-Data Cube(q) vocabulary[6] provide a description of the schema of a data set as well as its quantitative properties. Here, the RDF-Data Cube(q) vocabulary focuses specifically on aggregated data stored in data cubes, allowing to describe the cube's structure as well as the representation of the contained data. In contrast, the Discovery Vocabulary covers the description of raw data, but is not concerned with its representation. It also focuses mainly on file-based data sets.

The Disco vocabulary also makes use of DCMI Metadata Terms [7] for describing properties like temporal and spatial extend, or information like licenses. We therefore reuse the Disco vocabulary in DMS, to describe the schema and properties (including temporal, spatial, and license-related properties) of data sets.

[4] http://purl.org/spar
[5] http://www.sparontologies.net/ontologies/deo/source.html
[6] https://www.w3.org/TR/vocab-data-cube/
[7] http://dublincore.org/documents/dcmi-terms/

Ontologies for describing software: sciObjCS ontology [166] describes scientific objects (e.g. tools) along with their categories and creators. The OntoSoft ontology [67] is an OWL-based ontology for describing meta-data of scientific software. The ontology supports scientist to identify the software, and allows to cover many properties related to installing or running it. We deem the OntoSoft ontology complete; therefore, we exploit its classes and properties in DMS, to capture attributes such as dependencies, programming language, average runtime, etc.

Discussion. From our analysis of the state-of-the-art, we can conclude that existing ontologies already cover a subset of the identified requirements, but they all fall short covering the whole picture. Therefore, we aim at bridging this gap by combining relevant aspects of those ontologies, extending them with new concepts and properties. We also especially focus on the non-trivial relationships necessary for describing a pipeline, connecting data sets, methods, and software for a targeted usage context.

Ontology conceptualization

In this section, we will outline our conceptualisation of the DMS ontology. In addition to the ontologies listed in Table E.1, we build upon the PROV-O [8] ontology by extending the Entity PROV-O class for each of the classes of the DMS ontology that benefit from withholding provenance information (e.g. the creator, the location of a data set, etc). We further partially reused SKOS [9] ontology to make use of the taxonomy concepts (e.g. broader, narrower).

Figure E.3 provides a high-level, abstract view of the DMS ontology. The core concepts of our ontology are *data sets*, *methods*, and *software*. *Publications* implicitly describe pipelines, usually as parts of different *experiments*.

The DMS ontology has been implemented using OWL 2 DL, and consists of 10 classes and 30 properties. Table E.2 summarises the novel classes and properties included in DMS.

In Figure E.4, as an example, we zoom in the set of properties describing a data set, which is based on the *logicalDataset* concept of the *disco* ontology. The general properties of a data set cover the creator, licence, scope, link to location etc. Each *logicalDataset* has some data files (*disco:datafile*), and multiple logical data sets form the final data set. For example, one experiment might refer to a JSON file that contains a specific set of Twitter messages. We can distinguish the ground truth data set used in the experiment, with the *dms:isGroundTruthData* property. The variables (i.e., schema attributes) contained in the data set (e.g longitude, latitude of the tweets) can be defined using *disco:variable*. Dependent variables used in the experiment can be distinguished using the *dms:isDependentVariable*. Each data set has a scope (*dms:scope*) (e.g social media) which can be linked to a concept (skos:concept). It also includes the description of the temporal and spatial coverage of the dataset, which can be described using *dctersm:temporal* and *dctersm:spatial*. The statistical properties of the data set can be described using the *disco:DescriptiveStatistics* concept. The data files and at-

[8] https://www.w3.org/TR/prov-o/

[9] http://www.w3.org/2008/05/skos

Dataset	Classes	Scope, Application
	Properties	hasApplication , hasScope, hasStatisticMeasurement, isDependentVariable isGroundTruth
Method	Classes	MethodImplementation, Parameter, Application, AcceptanceRange,
	Properties	createdDataset, createdVariable, usedDataset, usedVariable, hasImplementation, implementedIn, referenceObjective, hasAcceptanceRange produced, comparedWith, hasEndRange, hasStartRange, measurementType
Sofware	Classes	SoftwareConfiguration, Application
	Properties	createdDataset, createdVariable, usedDataset, usedVariable, referenceObjective, produced
Experiment	Classes	Publication, Experiment, Objective, ResearchQuestion
	Properties	describesExperiment, usedMethod, usedSoftware, hasFigure, hasTable, hasConfiguration,relatedTo, isSubGoalOf, isSubResearchQuestionOf, hasObjective, hasResearchQuestion, hasCaption,

Table E.2: Summary of the novel classes and properties included in DMS. Some classes and properties are related both to the Method and the Software class

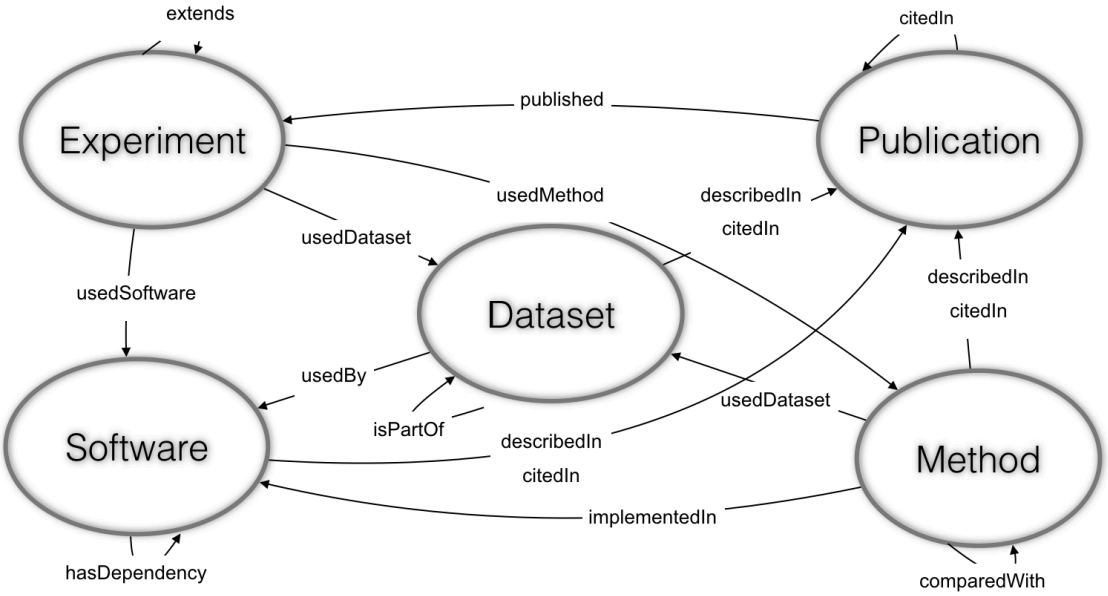


Figure E.3: Abstract view of the DMS ontology.

tributes used for each statistics can be described with the *disco:statisticsDatafile* and *disco:statisticsVariable* property.

Figure E.5 focuses on the parts of DMS that describe the link between data sets, methods, and software - this implicitly encodes the overall data processing pipeline discussed in a publication in the context of an experiment (as each experiment in the Big Data injection domain is usually sequence of applying methods to different data set, and assessing the result quality).

Here, an experiment has an objective (dms:objective) and uses data sets (dms:usedDataset), and methods (dms:usedMethod) as provided by software (dms:usedSoftware). In each experiment, different implementations or configurations of a method or software can be used, motivating *MethodImplementation* concept (dms:MethodImplementation $\subset$ deo:Methods). The application domain of the dataset, method or software can be described by the *dms:Application* class. Each application can be linked to the reference sub goal of the overall objective.

The dms:MethodImplementation, is either a new method described in the paper or an existing one (i.e. referenced) which can be used both for the creation (dms:createdDatset) or the analysis of a dataset. For instance, as shown in Figure E.6, a method by using a datafile (dms:usedDataset), some variables (dms:usedVariable) of the dataset, and having some parameters with an acceptance range, it can produce (dms:produced) a result (e.g precision 70%) with a measurement type (e.g. precision). This is the same for the dms:softwareConfiguration ($\subset$ ontosoft:Software) class, which can be used both for the creation or the analysis of a dataset.

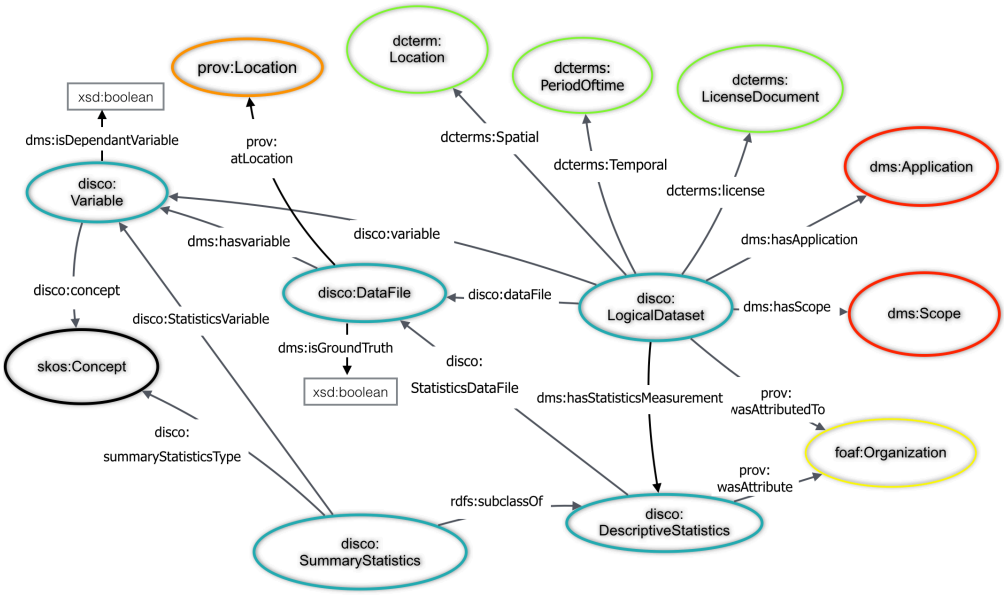


Figure E.4: The ovals with different colours used in the figure are an indicator of different ontologies.

E.4 Validation by Application

In this section we validate the suitability of DMS by a) manually annotating ten publications related to Social Media Big Data Injection, and b) providing example SPARQL queries to showcase that DMS can already satisfy many information demands identified by our industry practitioners in section E.2 - even without having complex reasoning capabilities in place which will be provided by future implementations.

Annotation of Scientific Publications

We manually annotated 10 papers [214, 47, 90, 112, 164, 218, 159, 157, 89, 194] in the field of social media analytics to show that our ontology is indeed able to capture the relevant properties of data processing pipelines. A public SPARQL endpoint to the RDF encoding resulting from this annotation is freely accessible[10]. Listing E.1 is a sample RDF representation of the annotation in Figure E.1: the *Police Shooting Data* has some schema attributes (called variables in accordance to the vocabulary of the DISCO ontology which we imported for this purpose) such as the cause of shooting incident, information on the victims liek age, gender, or race, and also the time and location of the shooting. 10,000 incident records have been collected during the period between 01/01/2000 to 15/06/2015, and the URL linking to location of the data set is http://www.fatalencounters.org. In Figure E.2, another example annotation is shown (from [214]).

[10] http://www.wis.ewi.tudelft.nl/DMS_SWM2017

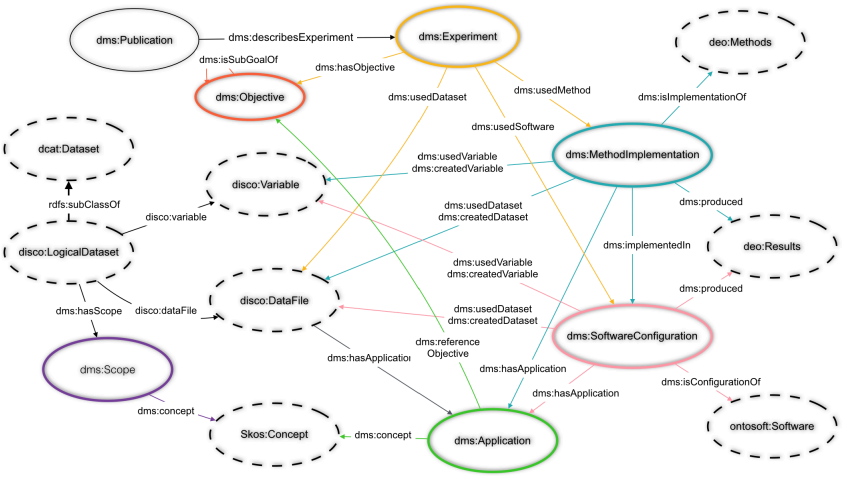


Figure E.5: Linking Datasets, Methods and Software. The ovals with different colors used in the figure are an indicator of different classes that we defined, and the dashed ovals are an indicator of old classes that were defined by the existing ontologies.

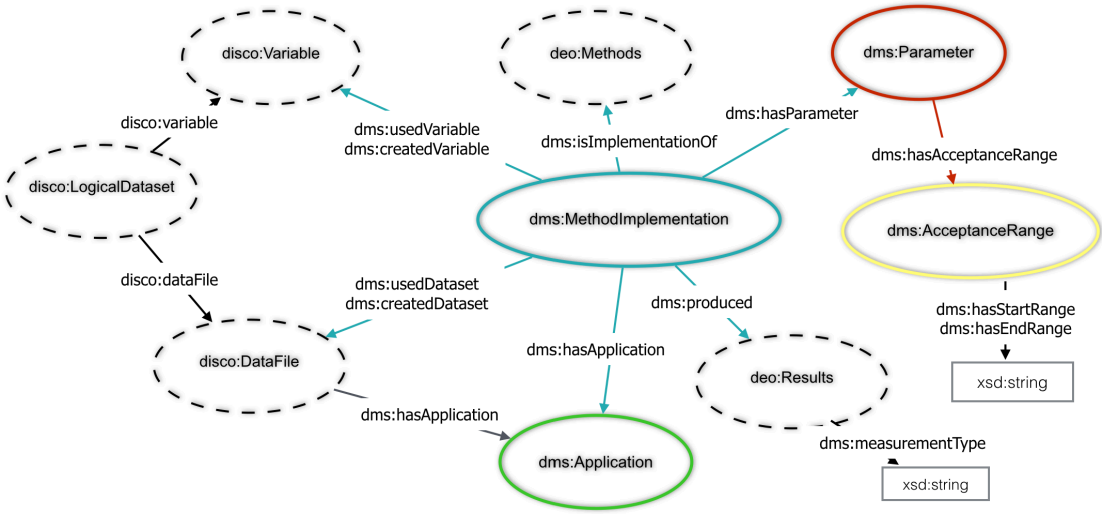


Figure E.6: A portion of ontology describing the method

The highlights represent information on methods, their application, used software and parameter, and the result of the overall pipeline. The resulting RDF is shown in Listing E.2.

```
prefix ns2:  <http://purl.org/dc/terms/> .
prefix ns1:  <http://www.w3.org/ns/prov> .
prefix ns4:  <http://purl.org/dc/elements/1.1/> .
prefix ns3:  <http://rdf-vocabulary.ddialliance.org/discovery> .
prefix ns5:  <https://github.com/mesbahs/DMS/blob/master/dms.owl> .
[ a              ns3:DataFile ;
```

```
  ns4:title        "Police Shooting Data" ;
  ns2:temporal     "01/01/2000-15/06/2015" ;
  ns3:caseQuantity 10000 ;
  ns5:hasVariable     [ a         ns3:Variable ;
                      ns1:value
                      "cause of police shooting incident" ] ;
  ns5:hasVariable     [ a         ns3:Variable ;
                      ns1:value  "time." ] ;
  ns5:hasVariable     [ a         ns3:Variable ;
                      ns1:value  "location" ] ;
  ns5:hasVariable     [ a         ns3:Variable ;
                      ns1:value
                      " race of the person being shot"  ] ;
  ns1:atLocation   "http://www.fatalencounters.org/"].
```

Listing E.1: RDF representation of Figure E.1.

```
prefix ns2:   <https://github.com/mesbahs/DMS/blob/master/dms.owl> .
prefix ns1:   <http://www.w3.org/ns/prov> .
prefix ns3:   <http://purl.org/dc/elements/1.1/> .

[ a                  ns2:methodImplementation ;
  ns3:title          "Lasso regularized regression" ;
  ns2:hasApplication [ a   ns2:Application ;
                       ns3:title  "modeling brand personality" ] ;
  ns2:hasParameter [ a  ns2:Parameter ;
                   ns1:value  "10-fold cross-validation"  ] ;
  ns2:produced [ a <http://purl.org/spar/deoResults> ;
                   ns2:measurementType  "predicted R2 "   ] ;
  ns2:implementedIn [ a  ns2:softwareConfiguration ;
                   ns3:title       "glmnet"    ;]
] .
```

Listing E.2: RDF representation of Figure E.2.

After manually annotating the 10 papers, we found that in general the ontology was able to cover the required properties related to data processing pipeline. As expected, some DMS concepts and properties were used more frequently than others, such as: Application, MethodImplementation, SoftwareConfiguration, produced, hasObjective, describesExperiment, usedDataset, usedMethod, usedSoftware, hasApplication et. On the other hand, some concepts were rarely used such as Parameter, AcceptanceRange, ResearchQuestion, Scope, etc.

Use Case Queries

In this section we will describe a sample query which the DMS ontology was designed to support. We envision that by populating the DMS ontology we are able to answer queries like the following examples:

Find the methods that can rank POI recommendation with a precision no lower than the state of the art

In this case, its related SPARQL query is as in Listing E.4. This query returns all the methods that can rank POI recommendation, and retrieves the ones that have the

highest precision among the state-of-the-art methods.

```
SELECT  ?method ?resvalue
    WHERE {
        ?method a dms:MethodImplementation;
            dms:produced ?result;
            dms:hasApplication ?application.
        ?application rdf:type dms:Application;
            disco:concept ?skosConcept.
            ?skosConcept rdf:type skos:Concept;
            skos:notation ?notation;
            FILTER (regex(?notation,"poi recommendation","i")).
        ?result rdf:type deo:Results;
            prov:value ?resvalue;
            Filter(?resvalue>= ?sota).
        ?result dms:measurementType ?type;
        FILTER(regex(?type,"precision","i")).
        { SELECT  ?sota
        WHERE {
           ?method a dms:MethodImplementation;
               dms:produced ?result;
               dms:hasApplication ?application.
            ?application disco:concept ?skosConcept.
               ?skosConcept  skos:notation ?notation;
               FILTER (regex(?notation,"poi recommendation","i")).
            ?result rdf:type deo:Results; prov:value ?sota;
               dms:measurementType ?type;
               FILTER(regex(?type,"precision","i")) }
           ORDER BY DESC(?sota) LIMIT 1}}}}
```

E.5 Conclusion & Outlook

In this paper, we presented a novel ontology DMS for covering meta-data of data sets, methods, and software as parts of Big Data injection pipelines found in scientific publications. We have presented a rigid process for designing the ontology based on information needs of data science practitioners, and the input of seasoned academic data science researchers. We also reused existing ontologies and vocabularies whenever possible, thus limiting the overhead of adapting our new ontology, which finally covers 10 classes and 30 properties in OWL 2 DL.

Finally, we validated the ontology by using it to annotate ten publications from the area of Social Media injection, and showing SPARQL queries which can indeed cover the information need identified by the practitioners in the requirement elicitation phase. In conclusion, one of the most dominant use-cases for external data injection is the field of social media analytics, and thus the publications and experts we used for eliciting the requirements and evaluating the effectiveness of DMS are rooted in that field. As a result, we believe that DMS is able to cover publications in that area well, but might need additional considerations when transferred to other domains.

Also, we aimed at a more generalized conceptualisation of the ontology due to the diversity of methods, data sets, measurement types, applications of methods, etc. While the chosen OWL 2 DL knowledge representation formalism would have allowed for a

more fine-grained model also including for example cardinalities or complex subclass taxonomies, we refrained from doing so with hindsight on the future usage of this ontology in a (semi-)automated information system. For example, the property "measurement type" could be further specialized into "precision", "recall", etc - but we felt that this would unnecessarily complicate the ontology. This argumentation is also in line with the minimal encoding bias and extendability principle guidelines outlined in [77].

In our future work, we will focus on realizing a larger ecosystem where the DMS ontology is semi-automatically populated using publications from digital library backend, thus building a rich knowledge repository of data processing pipeline meta-data which can serve as a nucleus for fostering future research on Big Data injection. This endeavour requires identifying rhetoric mentions of the properties and concepts covered in DMS in publications, not unlike the annotation task performed by our experts in the requirement elicitation step of this paper. This will likely motivate an expansion of DMS to also cover such rhetorical mentions on higher level of granularity which will be designed for human consumption, or as input for later processing steps (e.g. a natural language description of data set properties instead of a fine-grained explicit notion of the properties as used in this work). A second major challenge is realizing the reasoning capabilities necessary to support the queries identified during requirements analysis more effectively. For example, in the previous section, we implemented the notion of "current state-of-the-art" manually in SPARQL while in future versions of the system, such concepts should be usable without explicit definition.

Appendix F

Concept Focus: Semantic Meta-Data For Describing MOOC Content

This appendix contains an example of the application of the extracted long-tail entities in the MOOC domain.

Abstract

MOOCs promised to herald a new age of open education. However, efficient access to MOOC content is still hard, thus unnecessarily complicating many use cases like efficient re-use of material, or tailored access for life-long learning scenarios. One of the reasons for this lack of accessibility is the shortage of meaningful semantic meta-data describing MOOC content and the resulting learning experience. In this paper, we explore *Concept Focus*, a new type of meta-data for describing a perceptual facet of modern video-based MOOCs, capturing how focused a learning resource is topic-wise, which is often an indicator of clarity and understandability. We provide the theoretical foundations of *Concept Focus* and outline a methodical workflow of how to automatically compute it for MOOC lectures. Furthermore, we show that the learners' consumption behavior is correlated with a MOOC lecture's *Concept Focus*, thus underlining that this type of meta-data is indeed relevant for user-centric querying, personalizing or even designing the MOOC experience. For showing this, we performed an extensive study with real-life MOOCs and 12,849 learners over the duration of three months.

F.1 Introduction

Reusing and sharing teaching material is considered a central societal challenge by several policy makers. Despite continuously advancing open education policies [169], the vision of easy and personalizable access to open educational resources has still not been realized. To a large extent, this can be attributed to the lack of semantic capabilities of current courseware platforms: with access to only shallow *system-centric* meta-data (e.g. video

length, authors names, publication date), these platforms are mostly degraded to be simplistic repositories for storing and serving learning resources. As a result, such platforms are often lacking in usability [204], and rarely take advantage of emerging technologies as for example intelligent digital assistants or conversational interfaces [181]. In this paper, we advocate for the availability of semantic meta-data for educational resources. In contrast to *system-centric* meta-data, *semantic* meta-data – e.g. didactic intent, perceived difficulty, required expertise, or educational quality – describes the expected learning experience that a MOOC student might have with a given learning resource. This type of meta-data is generally hard to obtain as it either relies on subjective user-feedback, or needs to be indirectly approximated from the actual learning content. While some standards implicitly, introduce such meta-data types (e.g. LOM [36] – Learning Object Meta-data – covers "semantic density" or "difficulty"), it is usually not specified how such meta-data is defined, nor how it can be obtained from learning resources.

The main goal of this paper is to introduce the notion of **Concept Focus**, a measure of semantic relatedness of all concepts expressed in a learning resource. We set up a large-scale study on 3 MOOCs that engaged more than 12K learners over the duration of three months. We show that *Concept Focus*, while describing an intrinsic property of the learning resource, is also closely related to learner behavior patterns that are usually associated with difficulty or obstacles in the learning process. This can allow future work to use *Concept Focus* as a lever for learning personalization, e.g. steering certain types of learners towards content with high or low focus based on their personalities and learning goals. In summary, our original contributions include:

- The theoretical foundations for *Concept Focus*, a novel meta-data type capturing a relevant aspect of the learning experience of a MOOC video.
- The design space for methods that automatically obtain *Concept Focus* scores of a given MOOC video in a unsupervised fashion.
- The analysis of 3 real-life MOOC courses featuring 67 videos and 12,849 enrolled learners. We show that *Concept Focus* is a characterizing property of video scripts, describing their topical depth or width. We also report the presence of a significant correlation between *Concept Focus* scores and behavioral patterns indicating learning difficulties, e.g. video watching behaviour, quiz scores, and number of forum questions.

F.2 Concept Focus: Foundation and Implementation

Educational resources have been described by a multitude of different meta-data types, e.g. the IEEE LOM standard [38] includes a variety of different meta-data types, which can roughly be categorized into 9 groups. Most of these groups describe a learning object from a *system-centric* point of view: for example, general meta-data (e.g., id, title, language), technical aspects (e.g. length or size of videos), life-cycle (e.g., name of authors, version numbers), copyright, and usage restrictions. Only few types of meta-data actually cover the content itself: for instance, LOM group "classification" describes topic and

keywords. Only one group of meta-data in LOM ("educational") is dedicated to *learners* and their actual *learning experience*, with information about interactivity, difficulty or semantic density. This is analogous to other educational meta-data standards, as for example Ariadne [54]. Additionally, also bottom up approaches employing folksonomy techniques have emerged [26], with *educational* meta-data related to topical depth and didactic purpose being of central importance there.

This *educational* meta-data has been shown to be very beneficial for personalization and querying (especially data on difficulty, interactivity and density [156]), and its effectiveness even increases when combined with content-related meta-data [2]. Despite this fact, educational meta-data is rarely used in real-life MOOC systems. This can be attributed to the fact that it is expensive to obtain, and usually either expert judgments or crowd-sourcing needs to be employed to this end [156]. Furthermore, in [52], it has been shown that for effective personalization, more semantically deeper types (like learning styles or content properties) are beneficial, as they would allow for more meaningful similarity measurements between learning resources [52] for recommendations and explorative queries. Also Concept Focus could be used to that end, allowing to distinguish broader lectures from topically narrower ones.

Intuition

We define *Concept Focus* as a measure of semantic relatedness of all concepts expressed in a learning resource (e.g. a recorded lecture, or a script). Intuitively, *Concept Focus* characterizes how strongly a learning resource focuses on a specific topic: Concept Focus is *high* when the concepts of a resource share topical affinity – e.g., a lecture on natural language processing, which discusses a technique like "word embeddings" is implemented, mentioning only related NLP techniques and mathematical concepts.

We will test in our evaluation the hypothesis that learning material covering different topics, possibly loosely related, lead to learning difficulties. Even in cases where low *Concept Focus* does not always lead to confusion and learning problems (as it might also characterize material giving summaries or overviews), we argue that it is in either case a valuable meta-data field to be considered by an educational personalizing information system, as we will show, it drives behaviours similar to the ones of meta-data that are harder to obtain, as for example clarity or difficulty. *Concept Focus* can be computed automatically by relying on a combination of NLP and information extraction techniques, thus overcoming the aforementioned limitation of prohibitively high costs of crowd-sourcing or expert feedback. In short, *Concept Focus* can be realized as follows:

1. Extract all concepts (i.e., filtered named entities) from the textual representation of a given learning object.

2. Measure the *Semantic Relatedness* of a given concept in the learning resource, w.r.t. all other concepts in the same resource.

3. Calculate the *Concept Focus* of a resource, as a function of the semantic relatedness of all the concepts therein contained. Intuitively, if all concepts are semantically closely related, the *Concept Focus* focus of the resource is high; or low, otherwise.

Concept Extraction

In the following, we discuss how to extract concepts from videos, or more precisely the textual scripts of lecture videos. Arguably, the most important educational material in MOOCs are the videos, as they are the principal mean for content delivery.

They are therefore our main object of analysis. Due to their interactivity, videos have the additional benefit of enabling in-video interaction analysis (i.e. users click actions such as pauses, replaying, etc) to observe and assess the learning status of the students (e.g. difficulty in understanding the content) [118]. We exploit this fact in our evaluation.

Formally, a concept c can be defined as a k-gram that represents ideas and entities expressed in the video transcript text (e.g "machine learning", "stock price index") [168]. Automatic concept extraction from text has received much attention in the past decade [151, 20, 179, 143, 147], and thus there exist a number of publicly available concept extractor tools, relying on techniques such as term-frequency analysis [179], co-occurrence graph [151], etc. Extracting concepts from MOOCs content is, however, a challenging task due to the low-frequency problem [167]: MOOCs videos are relatively short documents and due to the small number of words, statistical techniques (e.g. co-occurence) are not applicable. To cater for such limitations, we employ an ensemble approach, running a battery of concept extractor tools on a video's script, and extracting all the concepts contained in it. We adopt:

- **TF-IDF**[1]: A well-know Information Retrieval technique, used to rank candidate concepts based on their tf-idf (term frequency - inverse document frequency) in the corpus.
- **TextRank [151]**: A technique that extracts concepts by ranking them according to their co-occurrence graph.
- **TopicRank [20]:** An extension of Textrank. A graph-based concept extraction approach which relies on a topical representation of the text.
- **KPMiner [56]**: A simple technique, which employs a set of heuristic rules (e.g. length of the concept, position in the sentence) to extract concepts from the text.
- **Rake [179]**: Rapid Automatic Keyword Extraction is able to identify concepts by relying on the term frequency, term degree, and ratio of degree to frequency.
- **TextRazor**[2]: A text analysis API that returns detected entities, possibly decorated with links to the DBpedia or Freebase knowledge bases.

[1] http://www.hlt.utdallas.edu/~saidul/code.html

[2] https://www.textrazor.com/

As a next step, we merge all the concepts individually extracted from each tool, filtering stopwords (e.g. something, anything, etc) and concepts coming from "common" English language (e.g., "events", "data") that could be found in Wordnet. We retain only concepts that have been detected by the majority of the extractor tools (i.e. 4 out of 6) to filter out irrelevant concepts (e.g. "six months", "new stories"). Intuitively, a concept will be considered as a correct concept if it has been harvested by different combinations of concept extraction tools [30]. By merging all concepts extracted from a given video scripts v, we obtain a final list of Candidate Concepts $concepts(v) = \{c_1, ..., c_N\}$.

Concept Focus

Concept Focus relies on measuring and aggregating the semantic relatedness of concepts contained in a lecture transcript: the higher the semantic relatedness between all concepts, the higher the focus of the lecture. While there can be many implementations for capturing semantic relatedness, previous studies [127] have shown that word embeddings [153] perform this task particularly well by e.g. measuring the cosine similarity of the word embedding vectors. We exploit Wikipedia to learn the word embedding representation of each concept. We first extract English articles from the latest publicly available Wikipedia dump[3]. Next, we built an embedding lexicon based on *fastText* [18]. *FastText* embeds each term (uni-gram and bi-gram) of a large document corpus into low-dimensional vector space (100 dimensions in our case) and overcomes the problem of out-of-vocabulary words by representing each word as a bag of character n-grams.

We adopt a typical measure of semantic relatedness $SR(c_1, c_2)$, that is computed between two specific concepts c_1 and c_2 by measuring the cosine similarity of their word embedding vectors [127].

In addition, we now also introduce the semantic relatedness $SR(c, v)$ between a concept c and all other concepts contained in a video transcript v. We also value the relatedness to the title of a video. For instance in a video v about "Propensity score matching"[4], concepts such as *propensity score*, *p-value* and *paired t-test* will get a higher semantic relatedness measure with respect to v, while a concept like *heart catheterization* is less related within v.

We define $SR(c, v)$ for a concept c and a MOOC video transcript v as follows:

$$SR(c, v) = \frac{\sum_{cv \in concepts(v)} SR(c, cv) * SR(c, titleOf(v))}{|concepts(v)|} \tag{F.1}$$

SR is a value in $[0, 1]$, where 1 represents the maximum relatedness that a concept can have in a video.

Consequently, the Concept Focus of a given lecture video v can be defined as the average concept relatedness of each concept in v within the context of v, i.e.:

[3] https://dumps.wikimedia.org/enwiki/20180201/

[4] https://www.coursera.org/learn/crash-course-in-causality/lecture/VtFdu/propensity-score-matching-in-r

$$CF(v) = \frac{\sum_{c \in concepts(v)} SR(c, v)}{|concepts(v)|} \tag{F.2}$$

CF is also in $[0, 1]$, where 1 is the highest *Concept Focus* value.

F.3 Evaluation

This section reports the results of an extensive study on real-life MOOCs, to showcase and discuss our new *Concept Focus* meta-data. We organize the study around the following research questions:

- **RQ1**: To what extent do properties of video scripts affect a course's *Concept Focus*? We investigate properties of learning material like video length, number of concepts, and position of the course in the MOOC.
- **RQ2**: To what extent does *Concept Focus* affect students' learning behaviour? We investigate the learners video watching behavior, quiz performance, and discussion behavior in relation to the Concept Focus of their consumed learning material.

Dataset Description

We analyze the log traces of learners collected from three MOOCs in edX[5]: itemize **DA** Data Analysis: Visualization and Dashboard Design, **IWC** Introduction to Water and Climate, **IWT** Introduction to Water Treatment.

We selected these 3 MOOCs for the following reasons: 1) they feature comparable amount of videos, and engaged students; 2) they cover a variety of topics; and 3) the scripts of their videos, and the interaction data for the engaged students are available. Table F.1 summarizes the main properties of the selected MOOCs. We consider only *engaged* learners, i.e. learners that watched at least one video for more than 15 seconds. Interaction data is collected through click log traces. We analyzed in total 9,899,369 log trace records of 12,849 learners. Statistics of the MOOC and learners are summarized in Table F.1.

[5]https://www.edx.org/

Table F.1: Overview of the Three MOOC datasets analyzed. Legend: REG – Registered; Eng – Engaged; CR – Completion Rate

					# Learners		
ID	**Name**	**Start**	**End**	**Videos**	**REG**	**ENG**	**CR**
DA	*Data Analysis*	03/2016	06/2016	22	32,682	5,711	3.74%
IWC	*Introduction Water and Climate*	09/2014	11/2014	27	9,267	4,947	2.60%
IWT	*Introduction Water Treatment*	01/2016	03/2016	18	13,198	2,191	3.07%

Properties of MOOCs. To answer **RQ1**, we study the relation between the following features of videos in a MOOC, and their *Concept Focus*:

- VD – *Video Duration*: the length of a video, expressed in seconds.
- VL – *Average Video Length*: the average number of words in the video scripts of the given MOOC.
- ANC – *Average Number of Concepts*: the average number of concepts extracted from the video scripts of the given MOOC.
- SC – *Session of the Course*: the date the lecture was given (i.e. first session, second session, etc)

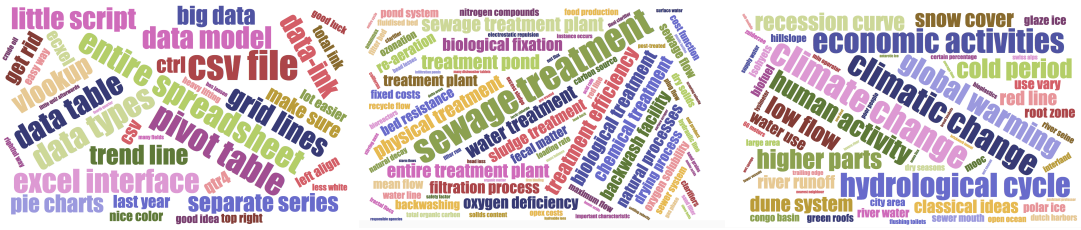
Learners Behaviour. To address **RQ2**, we study the relationship between the measured behaviour of learners, and the *Concept Focus* score of videos. From the log traces, we extracted the following 7 features. Each feature is calculated by aggregating all learner activities, including activities in the video player and in the course's forum, and their proficiency with the subject as assessed by the MOOC's grading system.

- WT – *Watching Time* of video material: the amount of time a learner has spent watching a video's material in the MOOC.
- NWT – *Normalized Watching Time* of video material: the total amount of time a learner has spent watching video material in the MOOC divided by the duration of the video.
- FS – # *Forward Seek*: the total number of times a learner seeks forward while watching a video.
- BS – # *Backward Seek*: the total number of times a learner seeks backward while watching a video.
- SU – # *Speed Up*: the total number of times a learner increases the play speed while watching a video.
- SD – # *Speed Down*: the total number of times a learner decrease the play speed while watching a video.
- FG – *Final Grade*: the percentage of quiz questions the learner. answered correctly after having interaction with a video.
- NFP – # *New Forum Posts*: the number of new forum posts (i.e., questions) created by the learner after having interaction with a video. Here we consider posts created within 15 minutes from the last interaction with a video.

RQ1: Video Properties vs. Concept Focus

Table F.2 summarizes the properties of the video scripts part of our analysis, including the number of unique concepts extracted from the MOOCs, the average, median and

standard deviation number of concepts extracted from their videos, as well as the length of the videos in terms of the number of words. Here we consider extracted concepts that were also present in Wikipedia, and for which a vector representation exists. Notably, 98% of the candidate concepts extracted from the concept extraction phase have a vector representation in our corpus. Figure F.1 shows samples of extracted concepts organized in word clouds, where the size of the concept is proportional to their Semantic Relatedness (SR) score.



(a) DA (b) IWT (c) IWC

Figure F.1: Extracted concepts from video scripts of IWC, IWT and DA.

DA videos, compared to IWC and IWT, feature on average 60% less concepts, and half the number of words per video. The standard deviation is proportionally higher, thus showing more variability within the course. Figure F.2 shows the distribution of the *Concept Focus* for all the videos of the three MOOCs. The average *Concept Focus* for the courses are respectively 0.29 for DA, 0.26 for IWT, and 0.19 for IWC. An example of IWC video with low focus score ($CF = 0.16$) is the lecture "Urban Engineering", [6] which includes a rather diverse concepts such as "cloaca maxima", "city wall", or "permeable pavements". The lecture belongs to introductory course on Water Climate, a subject that is bound to embrace several topics. The "Solver" lecture in the DA course[7] is an example of very focused video ($CF = 0.36$), including concepts such as "data table", "excel sheet", or "spreadsheet". This is also expected, as the lecture is exclusively about an Excel plug-in program called "Solver".

Figure F.3 shows the relation between the length of the video (in terms of words) and the *Concept Focus* for each MOOC. Intuitively, one would argue that the longer the text of the video script, the higher the number of concepts contained in it, thus the lower Concept Focus. Indeed, this is not necessarily the case. We can find a moderate significant positive correlation only for videos in the IWC course (Figure F.3c: $\rho = -0.59$, $p - value$: 0.0069). However, as shown in Figure F.4, videos with higher number of concepts do have lower concepts focus, but only for the DA course a moderate significant negative correlation could be found (Figure F.4a: $\rho = -0.60$, $p - value$: 0.01). These results show that *Concept Focus* is a lecture-specific property that is not biased by the length of a video or by the sheer number of concepts contained in it. Arguably, this is a desirable properties for a content-centric meta-data.

[6] https://www.youtube.com/watch?v=nhMcB-bwSF0

[7] https://www.youtube.com/watch?v=DgYmpmwBybQ

MOOC ID	UC	μC	mC	σC	μW	mW	σW
DA	298	17	16	6	680	624	262
IWT	687	49	46	12	1268	1303	365
IWC	1095	43	43	7	1481	1398	366

Table F.2: Descriptive statistics for concepts C and number of words W of the analyzed MOOCs video scripts. Legend: UC, Unique Concepts; μ, average; m, median; σ, std.

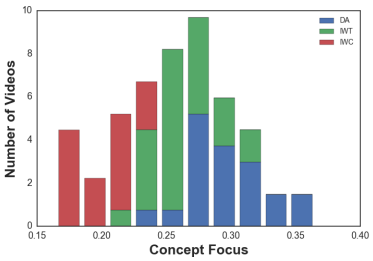
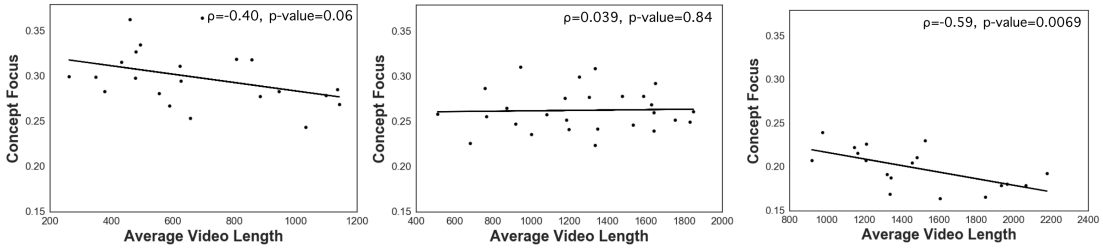


Figure F.2: Distribution of *Concept Focus* for the videos of *IWC*, *IWT* and *DA* in the shape of a stacked histogram

Finally, we study if the position of a video in a MOOC can be related to *Concept Focus*. Courses might feature different progression and organization of subject, with introductory lecture in the beginning (low *Concept Focus*) and specialized lectures later on (high *Concept Focus*). As shown in Figure F.5, the three courses feature very different teaching profiles. Despite the lack of statistically significant relation with *Concept Focus*, we can see how DA, for instance, starts with two very focused videos while, over time, lectures show consistent variations of *Concept Focus* scores. In IWT and IWC, on the other hand, the first lecture has low *Concept Focus*, and there is less variations in score across lectures, roughly remaining the same.



(a) DA $coeff = -4.645$, $intercept = 0.330$

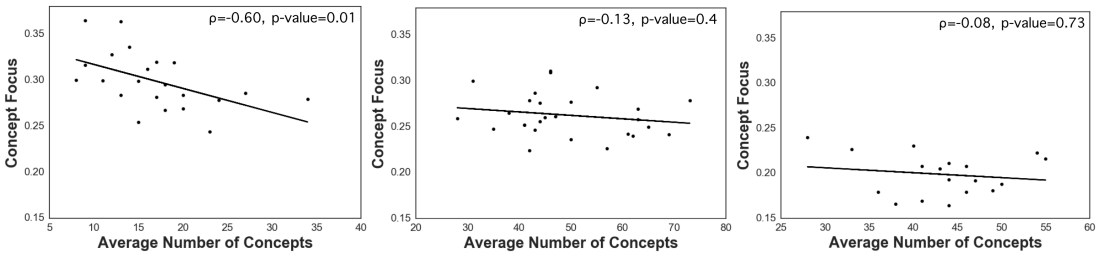
(b) IWT $coeff = 2.112$, $intercept = 0.2604$

(c) IWC $coeff = -3.763$, $intercept = 0.2549$

Figure F.3: *Concept Focus* and the number of words in the video transcripts

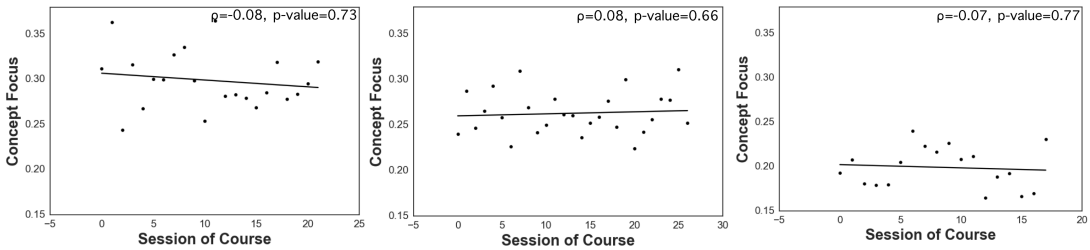
RQ2: Learning Behaviour vs. Concept Focus

We first study how the length of a video is related to the behaviour of learners, Figure F.6 summarizes the Spearman correlation between all measures as a heatmap. The *Video Duration* VD is obviously highly correlated with the learners *Watching Time* VT. The longer learners spends time watching videos, the higher the amount of video interactions such as FS (*# Forward Seek*), SU (*# Speed Up*) and SD (*# Speed Down*). We believe that the



(a) DA $coeff = -0.0026$, $intercept = 0.3430$ (b) WT $coeff = -0.0003$, $intercept = 0.28$ (c) IWC $coeff = -0.0005$, $intercept = 0.22$

Figure F.4: *Concept Focus* and the average number of concepts in video transcripts



(a) DA $coeff = -0.0007$, $intercept = 0.307$ (b) IWT $coeff = -0.00022$, $intercept = 0.2602$ (c) IWC $coeff = -0.0003$, $intercept = 0.2023$

Figure F.5: *Concept Focus* and the position of the related video in the MOOC.

	ρ	
NWT – # Normalized Watching Time	0.44	**
FS – # Forward Seek	0.31	*
BS – # Backward Seek	0.50	**
SU – # Speed Up	-0.36	**
SD – # Speed Down	-0.55	**
FG –Final Grade	0.19	
NFP – # New Forum Posts	-0.25	*

Table F.3: Spearman correlation ρ between *Concept Focus* and learners behavioural features for all the videos in the dataset. $*p - value < 0.05$, $**p - value < 0.001$

high WT is not associated with learning difficulty, as we observe a negative correlation between WT and BS, and positive correlation with SD which are indicators of higher level of difficulty [119].

Table F.3 reports the measured Spearman correlation between the *learners behaviour* metrics and *Concept Focus* of the corresponding videos. *Concept Focus* is significantly correlated with NWT, BS, SU, SD, and NFP. We observe a moderate positive correlation between the amount of time learners spent watching video lectures and the number of times they seek backward - i.e., in the videos with higher Concept Focus, learners watch the video

for a longer time and are more likely to re-watch parts of them. This observation aligns with the previous study [208] were the authors showed that difficulty correlates negatively with dwelling time (i.e time students spend watching a video). We interpret this result as a sign of students disengaging with videos having lower focus i.e. that cover a wider range of concepts. A similar result can be found in [103] where it has been shown that many students stop engaging with a courses (e.g. watching the videos) when they haven't enough knowledge to understand the context.

We also observe a weak negative correlation with the number of new forum post - i.e., after watching videos with lower Concept Focus, learners are more likely to post in the forum. This can be an indicator of having difficulty understanding the concepts in video scripts with low focus. The number of times the learner speed up and down the video have also a significant moderate negative correlation with the Concept Focus - i.e., in the videos with higher Concept Focus, learners continue watching the video without changing the speed of the video, possibly a sign of well-designed content progression. Finally we do not observe any statistically significant correlations between the final grade of the students and the Concept Focus.

The box plots in Figure F.7 depict the break down of the distribution of final grade, normalized watching time, # of new forum post, # of forward seek, # backward seek, # speed up and # speed down of three courses. In order to check if the samples are drawn from different population groups we performed the Kruskal-Wallis H Test (KWHT). In DA, where average *Concept Focus* is higher (0.29) than IWT (0.26) and IWC (0.19), the learners achieve a slightly higher grade (KWHT $statistic = 5.99$, $pvalue = 0.049$); a statistically significant higher normalized watching time (KWHT $statistic = 26.73$, $p - value = 1.56e - 06$), forward seek (KWHT $statistic = 10.49$, $pvalue = 0.005$) and back ward seek (KWHT $statistic = 17.31$, $pvalue = 0.0001$); and slightly lower number of speed up (KWHT $statistic = 9.94$, $pvalue = 0.006$) and speed down (KWHT $statistic = 1.35e - 05$, $pvalue = 22.42e - 05$). The difference in the distribution of number of new forum posts is not statistically significant (KWHT $statistic = 5.16$,

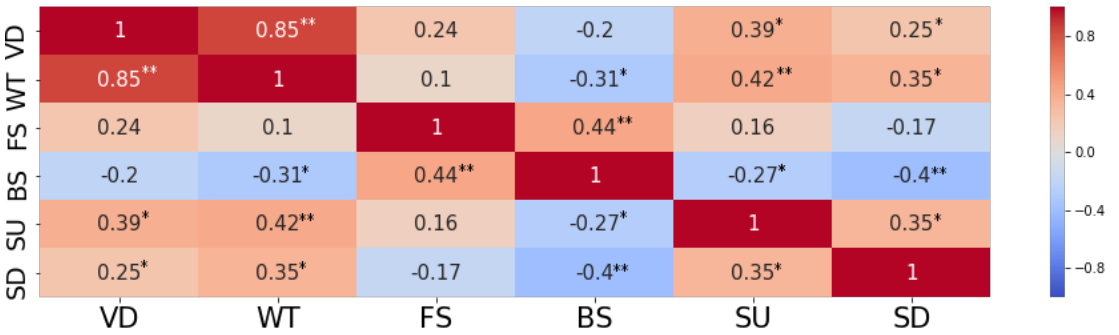


Figure F.6: Correlation heatmap of video interaction. Legend: *VD* – Video Duration; *WT* – Watching Time; *FS* – Forward Seeks; *BS* – Backward Seeks; *SU* – Speed Ups; *SD* – Speed Downs. $*p - value < 0.05$, $**p - value < 0.001$

$pvalue = 0.07$).

Altogether, these results show that *Concept Focus* is indeed a measure that relates to user-centric properties of videos, giving insights into potential engagement of learners, types of content, or potential learning problems.

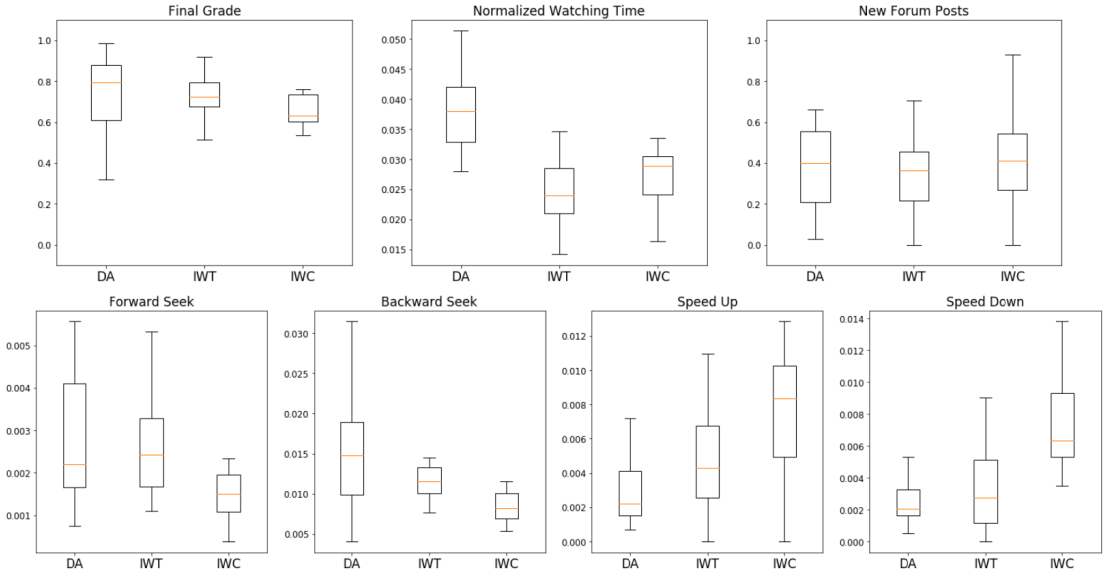


Figure F.7: Distribution of Final Grade (FG), Normalized Watching Time (NWT), # New Forum Posts (NFP), # Forward Seek (FS), # Backward Seek (BS), # Speed Up (SU) and # Speed Down (SD) for the three courses.

F.4 Related Work

A growing body of literature has examined different attributes (e.g. video length [78], interface characteristics [105], video textual complexity [208], displaying the instructor's face to video instruction[107]) of MOOC videos and their effect on learners' dwelling time [208, 119, 118] or dropout [78].

Recently, several studies focused on the in-video interactions analysis (e.g. measuring the number of pauses, skipping, re-watching) to measure the level of the perceived video difficulty [119, 118] and to model students learning behaviour [208]. The existing research capitalize on the relationship between the user and the content to measure the perceived video difficulty. We still have a limited understanding about the intrinsic properties of the text (i.e without the interpretation of the users) that make a MOOC video clear for the students. Our work is inspired by [208], where the researchers focused on the textual analysis (e.g. word and sentence length, frequency of words, etc) of the video scripts and showed the effect of video complexity on the users video interaction (i.e. dwelling time and rate of the learners). However, the properties of the concepts (i.e. k-grams that represent ideas and entities expressed in the text such as: machine learning, stock

price index, etc.) used in the text and the semantic relation between them are not well understood to characterize the lecture clarity and understandability. Thus, in this paper we focus on analyzing the content of MOOC videos to obtain their concept focus topic-wise, which is often an indicator of clarity and understandability of a lecture.

F.5 Conclusion

In this paper, we introduced *Concept Focus*, a novel type of meta-data capturing an aspect of a user's learning experience when interacting with learning content in an online MOOC platform. *Concept Focus* describes how focused a learning resource is w.r.t. a restricted set of topics. It can be used to semantically characterize a learning resource (as for example an in-depth explanations vs. a general overview), but might also be an indicator for potential learning challenges. In contrast to other meta-data types, we show that *Concept Focus* can be computed fully automatically by relying on a combination of natural language processing and information extraction techniques, thus avoiding the common detriment of having to rely on costly crowd-sourcing or experts. We believe Concept Focus can play a role as part of the feature set of more elaborate methods for automatically deriving meta-data on teaching methods or learning styles.

We conducted an extensive study covering three real-life MOOCs with 67 videos on the edX MOOC platform. We show that *Concept Focus* is a property that does not depend on video length, it is lecture-specific, and it characterizes the organization of a MOOC. By analyzing the activity logs of 12,849 learners, we investigated their video watching behavior, quiz performance, and discussion behavior in relation to the concept focus of their consumed learning material. Furthermore, we investigated properties of learning material like video length or number of contained concepts. The analysis indicates a correlation between low *Concept Focus*, and behaviors which are associated with learning difficulties.

While these results are supported by general intuition and previous findings, our study is limited to three MOOCs. Additional studies are therefore necessary to better understand the relationship between this novel meta-data, and behavioural properties of learners.

Summary

Named Entity Recognition (NER) is an essential information retrieval task. It enables a wide range of natural language processing applications such as semantic search, machine translation, etc. The NER can be formulated as the task of identifying and typing words or phrases in a text that refers to certain classes of interest (e.g., disease, Adverse Drug Reactions). There are different techniques to tackle NER, such as dictionary-based, rule-based, and machine learning-based. Machine learning-based NER techniques have shown to perform the best for entities with large amounts of human-labeled training datasets. However, their performance is limited when dealing with long-tail entities. Long-tail entities are entities that have a low frequency in the document collections and usually have no reference to existing Knowledge Bases. Obtaining human-labeled datasets is expensive and time-consuming, especially for long-tail entities that are scarcely available in document collections. This dissertation focuses on the problem of the lack of training data, arguably the largest bottleneck in training machine learning-based NER techniques. We investigated efficient and effective ways to augment training data by enhancing their size and quality automatically. Our work aimed at showing how, by enhancing the size and quality of the training data using different techniques, it will be possible to improve the performance of Long-tail Entity Recognition (L-tER).

The work is organised in four parts, each investigating a different training data augmentation technique to extract and type long-tail entities contained in scientific publications (Chapter 2, 3 and 4) and User Generated Content (Chapter 5). In Chapter 2, we use an existing pre-trained NER, which is trained on a large amount of training data, to check if it can be used for identifying the long-tail entities mentioned in the text. The results show that existing NER, being not trained for long-tail entity types, is not able to assign a label to the extracted entities. To tackle the problem of lack of training data for training a new NER, we then explore semantic expansion techniques (Chapter 3), generative models (Chapter 5) and a collaborative approach (Chapter 4) to augment the training data.

In Chapter 3, we describe a low-cost iterative approach for NER training called TSE-NER. We designed and evaluated a set of expansion strategies that exploit semantic similarity and relatedness between terms to expand on an initial set of data labeled with seed terms. We further presented several filtering heuristics to control the noise introduced by the expansion. Using this approach, we can tune the technique for either higher recall or higher precision scenarios. While promising, we observed that the heuristics are

prone to failure. We tackled this problem in Chapter 4 by incrementally incorporating human feedback on the relevance of extracted entities into the training cycle. The core goal of this chapter was to study how far does human feedback confirm or conflict with TSE-NER heuristics and how does incorporating human feedback into the TSE-NER filtering step improve the overall performance with respect to precision, recall, and F-measures. Our results show that by intelligently incorporating user feedback it is possible to decrease the number of false positives (i.e., 85.5% for *Dataset* and 54% for *Method* of entities). However, to show the full potential of the proposed approach, the pipeline needs to be integrated into an existing production system, like a large scale digital library, to receive continuous feedback from the system's users.

In Chapter 5, we propose a technique for augmenting the training data using deep generative models. We hypothesize that by leveraging deep probabilistic modeling to capture the underlying data structure, we can generate new training samples resembling the subset of the corpus for which human annotation is available. Extensive experiments on Twitter and Reddit datasets demonstrate that our approach can reduce the need for training data (reduced by 75%) and improve the overall performance of the L-tER.

This dissertation shows the need for novel Named Entity Recognition approaches targeting long-tail entities. We contribute novel techniques for training data augmentation that are capable of improving the performance of the long-tail entity recognizer. While we consider our results promising, in Chapter 6.2 we identify several directions for further investigation.

Samenvatting

Named Entity Recognition (NER) is een belangrijk onderdeel van information retrieval. Het maakt een breed scala mogelijk aan toepassingen van natuurlijke taalverwerking zoals semantisch zoeken, automatisch vertalen etc. We kunnen NER formuleren als het proces van het identificeren en typeren van woorden en zinnen in een tekst die verwijst naar specifieke klassen (bijv. ziekten, ongewenste reacties op medicijnen). Er zijn verschillende technieken voor NER, zoals technieken gebaseerd op woordenboeken, op regels, of op machine learning. Van NER-technieken gebaseerd op machine learning is aangetoond dat ze het beste werken voor entiteiten waarvoor grote hoeveelheden door mensen geannoteerde datasets beschikbaar zijn voor training. Maar hun performance is beperkt voor entiteiten uit de 'long tail'. 'Long tail'-entiteiten zijn entiteiten die in lage frequentie voorkomen in de documentcollecties en doorgaans geen verwijzing kennen naar bestaande kennisbanken. Het verkrijgen van door mensen geannoteerde datasets is duur en tijdrovend, in het bijzonder voor long tail-entiteiten die weinig voorkomen in documentcollecties. Dit proefschrift richt zich op het probleem van het gebrek aan trainingsdata, de grootste bottleneck in het trainen van op machine learning gebaseerde NER-technieken. We onderzoeken efficiënte en effectieve manieren om trainingsdata te verrijken door automatisch hun volume en kwaliteit te verbeteren. Ons werk richt zich op het aantonen hoe, door het volume en de kwaliteit van de trainingsdata met verschillende technieken te verbeteren, het mogelijk is om de performance te verbeteren van Long-tail Entity Recognition (L-tER).

Dit werk is verdeeld in vier delen, die elk een verschillende techniek onderzoeken om trainingsdata te verbeteren om long tail-entiteiten te extraheren en te typeren in wetenschappelijke publicaties (Hoofdstuk 2, 3 en 4) en user-generated content (Hoofdstuk 5). In Hoofdstuk 2 gebruiken we een bestaande vooraf getrainde NER-techniek, die getraind is op een grote hoeveelheid trainingsdata, om na te gaan of deze gebruikt kan worden voor het identificeren van long tail-entiteiten voorkomend in de tekst. De resultaten laten zien dat bestaande NER-technieken die niet getraind zijn voor long tail-entiteiten niet in staat zijn een label toe te kennen aan geëxtraheerde entiteiten. Om het probleem aan te pakken van het gebrek aan trainingsdata om een nieuwe NER-techniek te trainen, verkennen we technieken van semantische uitbreiding (Hoofdstuk 3), generatieve modellen (Hoofdstuk 5) en een collaboratieve aanpak (Hoofdstuk 4) om trainingsdata te verrijken.

In Hoofdstuk 3 beschrijven we een goedkope iteratieve aanpak om NRE te trainen

genaamd TSE-NER. We ontwerpen en evalueren een verzameling van uitbreidingsstrategieën die gebruik maken van semantische overeenkomst en verwantschap tussen termen om een dataset uit te breiden die gelabeld is met een aantal starttermen. Verder presenteren we verschillende filterheuristieken om de ruis van de uitbreiding te beheersen. Met deze aanpak kunnen we de techniek verfijnen voor scenarios waarin we meer of betere resultaten willen. Hoewel veelbelovend, observeren we dat de heuristieken vatbaar zijn voor fouten. We pakken dit probleem in Hoofdstuk 4 aan door stapje voor stapje gebruik te maken van menselijke feedback op de relevantie van geëxtraheerde entiteiten voor de trainingscyclus. Het kerndoel van dit hoofdstuk is om te bestuderen in hoeverre menselijke feedback TSE-NER-heuristieken bevestigt of tegenspreekt en hoe het meenemen van menselijke feedback in de TSE-NER-filterstap de totale performance verbetert met betrekking tot precisie, recall en F-measure. Onze resultaten laten zien dat door intelligent gebruikersfeedback mee te nemen, het mogelijk is om het aantal false positives (85% voor Dataset en 54% voor Method) te verminderen. Echter om de voorgestelde aanpak volledig te kunnen benutten moet de pijplijn geïntegreerd worden in een bestaand productiesysteem, zoals een grootschalige digitale bibliotheek, om continue feedback te ontvangen van de systeemgebruikers.

In Hoofdstuk 5 stellen we een techniek voor om trainingsdata te verbeteren met diepgeneratieve modellen. We veronderstellen dat door diep-probabilistisch modelleren te gebruiken om de onderliggende datastructuur te vatten, we nieuwe trainingsvoorbeelden kunnen genereren die lijken op de deelverzameling van het corpus waarvoor menselijke annotaties beschikbaar zijn. Uitgebreide experimenten op datasets van Twitter en Reddit laten zien dat onze aanpak de behoefte aan trainingsdata kan reduceren (met 75%) en de totale performance kan verbeteren van L-tER.

Dit proefschrift toont de behoefte aan nieuwe Named Entity Recognition-aanpakken gericht op long tail-entiteiten. Onze contributie bestaat uit nieuwe technieken om trainingsdata te verrijken die ons in staat stellen om de performance van het herkennen van long tail- entiteiten te verbeteren. We achten onze resultaten veelbelovend en geven in Hoofdstuk 6.2 enkele richtingen voor verder onderzoek.

SIKS Dissertation Series

Since 1998, all dissertations written by Ph.D. students who have conducted their research under auspices of a senior research fellow of the SIKS research school are published in the SIKS Dissertation Series.

2011 01 Botond Cseke (RUN), Variational Algorithms for Bayesian Inference in Latent Gaussian Models
02 Nick Tinnemeier (UU), Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language
03 Jan Martijn van der Werf (TUE), Compositional Design and Verification of Component-Based Information Systems
04 Hado van Hasselt (UU), Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference
05 Bas van der Raadt (VU), Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline.
06 Yiwen Wang (TUE), Semantically-Enhanced Recommendations in Cultural Heritage
07 Yujia Cao (UT), Multimodal Information Presentation for High Load Human Computer Interaction
08 Nieske Vergunst (UU), BDI-based Generation of Robust Task-Oriented Dialogues
09 Tim de Jong (OU), Contextualised Mobile Media for Learning
10 Bart Bogaert (UvT), Cloud Content Contention
11 Dhaval Vyas (UT), Designing for Awareness: An Experience-focused HCI Perspective
12 Carmen Bratosin (TUE), Grid Architecture for Distributed Process Mining
13 Xiaoyu Mao (UvT), Airport under Control. Multiagent Scheduling for Airport Ground Handling
14 Milan Lovric (EUR), Behavioral Finance and Agent-Based Artificial Markets
15 Marijn Koolen (UvA), The Meaning of Structure: the Value of Link Evidence for Information Retrieval
16 Maarten Schadd (UM), Selective Search in Games of Different Complexity
17 Jiyin He (UVA), Exploring Topic Structure: Coherence, Diversity and Relatedness
18 Mark Ponsen (UM), Strategic Decision-Making in complex games
19 Ellen Rusman (OU), The Mind's Eye on Personal Profiles

20 Qing Gu (VU), Guiding service-oriented software engineering - A view-based approach
21 Linda Terlouw (TUD), Modularization and Specification of Service-Oriented Systems
22 Junte Zhang (UVA), System Evaluation of Archival Description and Access
23 Wouter Weerkamp (UVA), Finding People and their Utterances in Social Media
24 Herwin van Welbergen (UT), Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior
25 Syed Waqar ul Qounain Jaffry (VU), Analysis and Validation of Models for Trust Dynamics
26 Matthijs Aart Pontier (VU), Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots
27 Aniel Bhulai (VU), Dynamic website optimization through autonomous management of design patterns
28 Rianne Kaptein (UVA), Effective Focused Retrieval by Exploiting Query Context and Document Structure
29 Faisal Kamiran (TUE), Discrimination-aware Classification
30 Egon van den Broek (UT), Affective Signal Processing (ASP): Unraveling the mystery of emotions
31 Ludo Waltman (EUR), Computational and Game-Theoretic Approaches for Modeling Bounded Rationality
32 Nees-Jan van Eck (EUR), Methodological Advances in Bibliometric Mapping of Science
33 Tom van der Weide (UU), Arguing to Motivate Decisions
34 Paolo Turrini (UU), Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations
35 Maaike Harbers (UU), Explaining Agent Behavior in Virtual Training
36 Erik van der Spek (UU), Experiments in serious game design: a cognitive approach
37 Adriana Burlutiu (RUN), Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference
38 Nyree Lemmens (UM), Bee-inspired Distributed Optimization
39 Joost Westra (UU), Organizing Adaptation using Agents in Serious Games
40 Viktor Clerc (VU), Architectural Knowledge Management in Global Software Development
41 Luan Ibraimi (UT), Cryptographically Enforced Distributed Data Access Control
42 Michal Sindlar (UU), Explaining Behavior through Mental State Attribution
43 Henk van der Schuur (UU), Process Improvement through Software Operation Knowledge
44 Boris Reuderink (UT), Robust Brain-Computer Interfaces
45 Herman Stehouwer (UvT), Statistical Language Models for Alternative Sequence Selection

46 Beibei Hu (TUD), Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work
47 Azizi Bin Ab Aziz (VU), Exploring Computational Models for Intelligent Support of Persons with Depression
48 Mark Ter Maat (UT), Response Selection and Turn-taking for a Sensitive Artificial Listening Agent
49 Andreea Niculescu (UT), Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality

---

2012 01 Terry Kakeeto (UvT), Relationship Marketing for SMEs in Uganda
02 Muhammad Umair (VU), Adaptivity, emotion, and Rationality in Human and Ambient Agent Models
03 Adam Vanya (VU), Supporting Architecture Evolution by Mining Software Repositories
04 Jurriaan Souer (UU), Development of Content Management System-based Web Applications
05 Marijn Plomp (UU), Maturing Interorganisational Information Systems
06 Wolfgang Reinhardt (OU), Awareness Support for Knowledge Workers in Research Networks
07 Rianne van Lambalgen (VU), When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions
08 Gerben de Vries (UVA), Kernel Methods for Vessel Trajectories
09 Ricardo Neisse (UT), Trust and Privacy Management Support for Context-Aware Service Platforms
10 David Smits (TUE), Towards a Generic Distributed Adaptive Hypermedia Environment
11 J.C.B. Rantham Prabhakara (TUE), Process Mining in the Large: Preprocessing, Discovery, and Diagnostics
12 Kees van der Sluijs (TUE), Model Driven Design and Data Integration in Semantic Web Information Systems
13 Suleman Shahid (UvT), Fun and Face: Exploring non-verbal expressions of emotion during playful interactions
14 Evgeny Knutov (TUE), Generic Adaptation Framework for Unifying Adaptive Web-based Systems
15 Natalie van der Wal (VU), Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes.
16 Fiemke Both (VU), Helping people by understanding them - Ambient Agents supporting task execution and depression treatment
17 Amal Elgammal (UvT), Towards a Comprehensive Framework for Business Process Compliance
18 Eltjo Poort (VU), Improving Solution Architecting Practices
19 Helen Schonenberg (TUE), What's Next? Operational Support for Business Process Execution
20 Ali Bahramisharif (RUN), Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing
21 Roberto Cornacchia (TUD), Querying Sparse Matrices for Information Retrieval

22 Thijs Vis (UvT), Intelligence, politie en veiligheidsdienst: verenigbare grootheden?
23 Christian Muehl (UT), Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction
24 Laurens van der Werff (UT), Evaluation of Noisy Transcripts for Spoken Document Retrieval
25 Silja Eckartz (UT), Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application
26 Emile de Maat (UVA), Making Sense of Legal Text
27 Hayrettin Gurkok (UT), Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games
28 Nancy Pascall (UvT), Engendering Technology Empowering Women
29 Almer Tigelaar (UT), Peer-to-Peer Information Retrieval
30 Alina Pommeranz (TUD), Designing Human-Centered Systems for Reflective Decision Making
31 Emily Bagarukayo (RUN), A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure
32 Wietske Visser (TUD), Qualitative multi-criteria preference representation and reasoning
33 Rory Sie (OUN), Coalitions in Cooperation Networks (COCOON)
34 Pavol Jancura (RUN), Evolutionary analysis in PPI networks and applications
35 Evert Haasdijk (VU), Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics
36 Denis Ssebugwawo (RUN), Analysis and Evaluation of Collaborative Modeling Processes
37 Agnes Nakakawa (RUN), A Collaboration Process for Enterprise Architecture Creation
38 Selmar Smit (VU), Parameter Tuning and Scientific Testing in Evolutionary Algorithms
39 Hassan Fatemi (UT), Risk-aware design of value and coordination networks
40 Agus Gunawan (UvT), Information Access for SMEs in Indonesia
41 Sebastian Kelle (OU), Game Design Patterns for Learning
42 Dominique Verpoorten (OU), Reflection Amplifiers in self-regulated Learning
43 Withdrawn
44 Anna Tordai (VU), On Combining Alignment Techniques
45 Benedikt Kratz (UvT), A Model and Language for Business-aware Transactions
46 Simon Carter (UVA), Exploration and Exploitation of Multilingual Data for Statistical Machine Translation
47 Manos Tsagkias (UVA), Mining Social Media: Tracking Content and Predicting Behavior
48 Jorn Bakker (TUE), Handling Abrupt Changes in Evolving Time-series Data
49 Michael Kaisers (UM), Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions
50 Steven van Kervel (TUD), Ontologogy driven Enterprise Information Systems Engineering

51 Jeroen de Jong (TUD), Heuristics in Dynamic Sceduling; a practical framework with a case study in elevator dispatching

---

2013 01 Viorel Milea (EUR), News Analytics for Financial Decision Support
02 Erietta Liarou (CWI), MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing
03 Szymon Klarman (VU), Reasoning with Contexts in Description Logics
04 Chetan Yadati (TUD), Coordinating autonomous planning and scheduling
05 Dulce Pumareja (UT), Groupware Requirements Evolutions Patterns
06 Romulo Goncalves (CWI), The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience
07 Giel van Lankveld (UvT), Quantifying Individual Player Differences
08 Robbert-Jan Merk (VU), Making enemies: cognitive modeling for opponent agents in fighter pilot simulators
09 Fabio Gori (RUN), Metagenomic Data Analysis: Computational Methods and Applications
10 Jeewanie Jayasinghe Arachchige (UvT), A Unified Modeling Framework for Service Design.
11 Evangelos Pournaras (TUD), Multi-level Reconfigurable Self-organization in Overlay Services
12 Marian Razavian (VU), Knowledge-driven Migration to Services
13 Mohammad Safiri (UT), Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly
14 Jafar Tanha (UVA), Ensemble Approaches to Semi-Supervised Learning Learning
15 Daniel Hennes (UM), Multiagent Learning - Dynamic Games and Applications
16 Eric Kok (UU), Exploring the practical benefits of argumentation in multi-agent deliberation
17 Koen Kok (VU), The PowerMatcher: Smart Coordination for the Smart Electricity Grid
18 Jeroen Janssens (UvT), Outlier Selection and One-Class Classification
19 Renze Steenhuizen (TUD), Coordinated Multi-Agent Planning and Scheduling
20 Katja Hofmann (UvA), Fast and Reliable Online Learning to Rank for Information Retrieval
21 Sander Wubben (UvT), Text-to-text generation by monolingual machine translation
22 Tom Claassen (RUN), Causal Discovery and Logic
23 Patricio de Alencar Silva (UvT), Value Activity Monitoring
24 Haitham Bou Ammar (UM), Automated Transfer in Reinforcement Learning
25 Agnieszka Anna Latoszek-Berendsen (UM), Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System
26 Alireza Zarghami (UT), Architectural Support for Dynamic Homecare Service Provisioning

27 Mohammad Huq (UT), Inference-based Framework Managing Data Provenance
28 Frans van der Sluis (UT), When Complexity becomes Interesting: An Inquiry into the Information eXperience
29 Iwan de Kok (UT), Listening Heads
30 Joyce Nakatumba (TUE), Resource-Aware Business Process Management: Analysis and Support
31 Dinh Khoa Nguyen (UvT), Blueprint Model and Language for Engineering Cloud Applications
32 Kamakshi Rajagopal (OUN), Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development
33 Qi Gao (TUD), User Modeling and Personalization in the Microblogging Sphere
34 Kien Tjin-Kam-Jet (UT), Distributed Deep Web Search
35 Abdallah El Ali (UvA), Minimal Mobile Human Computer Interaction
36 Than Lam Hoang (TUe), Pattern Mining in Data Streams
37 Dirk Börner (OUN), Ambient Learning Displays
38 Eelco den Heijer (VU), Autonomous Evolutionary Art
39 Joop de Jong (TUD), A Method for Enterprise Ontology based Design of Enterprise Information Systems
40 Pim Nijssen (UM), Monte-Carlo Tree Search for Multi-Player Games
41 Jochem Liem (UVA), Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning
42 Léon Planken (TUD), Algorithms for Simple Temporal Reasoning
43 Marc Bron (UVA), Exploration and Contextualization through Interaction and Concepts

---

2014 01 Nicola Barile (UU), Studies in Learning Monotone Models from Data
02 Fiona Tuliyano (RUN), Combining System Dynamics with a Domain Modeling Method
03 Sergio Raul Duarte Torres (UT), Information Retrieval for Children: Search Behavior and Solutions
04 Hanna Jochmann-Mannak (UT), Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation
05 Jurriaan van Reijsen (UU), Knowledge Perspectives on Advancing Dynamic Capability
06 Damian Tamburri (VU), Supporting Networked Software Development
07 Arya Adriansyah (TUE), Aligning Observed and Modeled Behavior
08 Samur Araujo (TUD), Data Integration over Distributed and Heterogeneous Data Endpoints
09 Philip Jackson (UvT), Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language
10 Ivan Salvador Razo Zapata (VU), Service Value Networks
11 Janneke van der Zwaan (TUD), An Empathic Virtual Buddy for Social Support

12 Willem van Willigen (VU), Look Ma, No Hands: Aspects of Autonomous Vehicle Control
13 Arlette van Wissen (VU), Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains
14 Yangyang Shi (TUD), Language Models With Meta-information
15 Natalya Mogles (VU), Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare
16 Krystyna Milian (VU), Supporting trial recruitment and design by automatically interpreting eligibility criteria
17 Kathrin Dentler (VU), Computing healthcare quality indicators automatically: Secondary Use of Patient Data and Semantic Interoperability
18 Mattijs Ghijsen (UVA), Methods and Models for the Design and Study of Dynamic Agent Organizations
19 Vinicius Ramos (TUE), Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support
20 Mena Habib (UT), Named Entity Extraction and Disambiguation for Informal Text: The Missing Link
21 Kassidy Clark (TUD), Negotiation and Monitoring in Open Environments
22 Marieke Peeters (UU), Personalized Educational Games - Developing agent-supported scenario-based training
23 Eleftherios Sidirourgos (UvA/CWI), Space Efficient Indexes for the Big Data Era
24 Davide Ceolin (VU), Trusting Semi-structured Web Data
25 Martijn Lappenschaar (RUN), New network models for the analysis of disease interaction
26 Tim Baarslag (TUD), What to Bid and When to Stop
27 Rui Jorge Almeida (EUR), Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty
28 Anna Chmielowiec (VU), Decentralized k-Clique Matching
29 Jaap Kabbedijk (UU), Variability in Multi-Tenant Enterprise Software
30 Peter de Cock (UvT), Anticipating Criminal Behaviour
31 Leo van Moergestel (UU), Agent Technology in Agile Multiparallel Manufacturing and Product Support
32 Naser Ayat (UvA), On Entity Resolution in Probabilistic Data
33 Tesfa Tegegne (RUN), Service Discovery in eHealth
34 Christina Manteli (VU), The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems.
35 Joost van Ooijen (UU), Cognitive Agents in Virtual Worlds: A Middleware Design Approach
36 Joos Buijs (TUE), Flexible Evolutionary Algorithms for Mining Structured Process Models
37 Maral Dadvar (UT), Experts and Machines United Against Cyberbullying
38 Danny Plass-Oude Bos (UT), Making brain-computer interfaces better: improving usability through post-processing.
39 Jasmina Maric (UvT), Web Communities, Immigration, and Social Capital

40 Walter Omona (RUN), A Framework for Knowledge Management Using ICT in Higher Education
41 Frederic Hogenboom (EUR), Automated Detection of Financial Events in News Text
42 Carsten Eijckhof (CWI/TUD), Contextual Multidimensional Relevance Models
43 Kevin Vlaanderen (UU), Supporting Process Improvement using Method Increments
44 Paulien Meesters (UvT), Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden.
45 Birgit Schmitz (OUN), Mobile Games for Learning: A Pattern-Based Approach
46 Ke Tao (TUD), Social Web Data Analytics: Relevance, Redundancy, Diversity
47 Shangsong Liang (UVA), Fusion and Diversification in Information Retrieval

2015 01 Niels Netten (UvA), Machine Learning for Relevance of Information in Crisis Response
02 Faiza Bukhsh (UvT), Smart auditing: Innovative Compliance Checking in Customs Controls
03 Twan van Laarhoven (RUN), Machine learning for network data
04 Howard Spoelstra (OUN), Collaborations in Open Learning Environments
05 Christoph Bösch (UT), Cryptographically Enforced Search Pattern Hiding
06 Farideh Heidari (TUD), Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes
07 Maria-Hendrike Peetz (UvA), Time-Aware Online Reputation Analysis
08 Jie Jiang (TUD), Organizational Compliance: An agent-based model for designing and evaluating organizational interactions
09 Randy Klaassen (UT), HCI Perspectives on Behavior Change Support Systems
10 Henry Hermans (OUN), OpenU: design of an integrated system to support lifelong learning
11 Yongming Luo (TUE), Designing algorithms for big graph datasets: A study of computing bisimulation and joins
12 Julie M. Birkholz (VU), Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks
13 Giuseppe Procaccianti (VU), Energy-Efficient Software
14 Bart van Straalen (UT), A cognitive approach to modeling bad news conversations
15 Klaas Andries de Graaf (VU), Ontology-based Software Architecture Documentation
16 Changyun Wei (UT), Cognitive Coordination for Cooperative Multi-Robot Teamwork
17 André van Cleeff (UT), Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs
18 Holger Pirk (CWI), Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories
19 Bernardo Tabuenca (OUN), Ubiquitous Technology for Lifelong Learners

20 Lois Vanhée (UU), Using Culture and Values to Support Flexible Coordination
21 Sibren Fetter (OUN), Using Peer-Support to Expand and Stabilize Online Learning
22 Zhemin Zhu (UT), Co-occurrence Rate Networks
23 Luit Gazendam (VU), Cataloguer Support in Cultural Heritage
24 Richard Berendsen (UVA), Finding People, Papers, and Posts: Vertical Search Algorithms and Evaluation
25 Steven Woudenberg (UU), Bayesian Tools for Early Disease Detection
26 Alexander Hogenboom (EUR), Sentiment Analysis of Text Guided by Semantics and Structure
27 Sándor Héman (CWI), Updating compressed colomn stores
28 Janet Bagorogoza (TiU), Knowledge Management and High Performance; The Uganda Financial Institutions Model for HPO
29 Hendrik Baier (UM), Monte-Carlo Tree Search Enhancements for One-Player and Two-Player Domains
30 Kiavash Bahreini (OU), Real-time Multimodal Emotion Recognition in E-Learning
31 Yakup Koç (TUD), On the robustness of Power Grids
32 Jerome Gard (UL), Corporate Venture Management in SMEs
33 Frederik Schadd (TUD), Ontology Mapping with Auxiliary Resources
34 Victor de Graaf (UT), Gesocial Recommender Systems
35 Jungxao Xu (TUD), Affective Body Language of Humanoid Robots: Perception and Effects in Human Robot Interaction

2016 01 Syed Saiden Abbas (RUN), Recognition of Shapes by Humans and Machines
02 Michiel Christiaan Meulendijk (UU), Optimizing medication reviews through decision support: prescribing a better pill to swallow
03 Maya Sappelli (RUN), Knowledge Work in Context: User Centered Knowledge Worker Support
04 Laurens Rietveld (VU), Publishing and Consuming Linked Data
05 Evgeny Sherkhonov (UVA), Expanded Acyclic Queries: Containment and an Application in Explaining Missing Answers
06 Michel Wilson (TUD), Robust scheduling in an uncertain environment
07 Jeroen de Man (VU), Measuring and modeling negative emotions for virtual training
08 Matje van de Camp (TiU), A Link to the Past: Constructing Historical Social Networks from Unstructured Data
09 Archana Nottamkandath (VU), Trusting Crowdsourced Information on Cultural Artefacts
10 George Karafotias (VUA), Parameter Control for Evolutionary Algorithms
11 Anne Schuth (UVA), Search Engines that Learn from Their Users
12 Max Knobbout (UU), Logics for Modelling and Verifying Normative Multi-Agent Systems
13 Nana Baah Gyan (VU), The Web, Speech Technologies and Rural Development in West Africa - An ICT4D Approach
14 Ravi Khadka (UU), Revisiting Legacy Software System Modernization

15 Steffen Michels (RUN), Hybrid Probabilistic Logics - Theoretical Aspects, Algorithms and Experiments
16 Guangliang Li (UVA), Socially Intelligent Autonomous Agents that Learn from Human Reward
17 Berend Weel (VU), Towards Embodied Evolution of Robot Organisms
18 Albert Meroño Peñuela (VU), Refining Statistical Data on the Web
19 Julia Efremova (Tu/e), Mining Social Structures from Genealogical Data
20 Daan Odijk (UVA), Context & Semantics in News & Web Search
21 Alejandro Moreno Célleri (UT), From Traditional to Interactive Playspaces: Automatic Analysis of Player Behavior in the Interactive Tag Playground
22 Grace Lewis (VU), Software Architecture Strategies for Cyber-Foraging Systems
23 Fei Cai (UVA), Query Auto Completion in Information Retrieval
24 Brend Wanders (UT), Repurposing and Probabilistic Integration of Data; An Iterative and data model independent approach
25 Julia Kiseleva (TU/e), Using Contextual Information to Understand Searching and Browsing Behavior
26 Dilhan Thilakarathne (VU), In or Out of Control: Exploring Computational Models to Study the Role of Human Awareness and Control in Behavioural Choices, with Applications in Aviation and Energy Management Domains
27 Wen Li (TUD), Understanding Geo-spatial Information on Social Media
28 Mingxin Zhang (TUD), Large-scale Agent-based Social Simulation - A study on epidemic prediction and control
29 Nicolas Höning (TUD), Peak reduction in decentralised electricity systems - Markets and prices for flexible planning
30 Ruud Mattheij (UvT), The Eyes Have It
31 Mohammad Khelghati (UT), Deep web content monitoring
32 Eelco Vriezekolk (UT), Assessing Telecommunication Service Availability Risks for Crisis Organisations
33 Peter Bloem (UVA), Single Sample Statistics, exercises in learning from just one example
34 Dennis Schunselaar (TUE), Configurable Process Trees: Elicitation, Analysis, and Enactment
35 Zhaochun Ren (UVA), Monitoring Social Media: Summarization, Classification and Recommendation
36 Daphne Karreman (UT), Beyond R2D2: The design of nonverbal interaction behavior optimized for robot-specific morphologies
37 Giovanni Sileno (UvA), Aligning Law and Action - a conceptual and computational inquiry
38 Andrea Minuto (UT), Materials that Matter - Smart Materials meet Art & Interaction Design
39 Merijn Bruijnes (UT), Believable Suspect Agents; Response and Interpersonal Style Selection for an Artificial Suspect
40 Christian Detweiler (TUD), Accounting for Values in Design
41 Thomas King (TUD), Governing Governance: A Formal Framework for Analysing Institutional Design and Enactment Governance

42 Spyros Martzoukos (UVA), Combinatorial and Compositional Aspects of Bilingual Aligned Corpora
43 Saskia Koldijk (RUN), Context-Aware Support for Stress Self-Management: From Theory to Practice
44 Thibault Sellam (UVA), Automatic Assistants for Database Exploration
45 Bram van de Laar (UT), Experiencing Brain-Computer Interface Control
46 Jorge Gallego Perez (UT), Robots to Make you Happy
47 Christina Weber (UL), Real-time foresight - Preparedness for dynamic innovation networks
48 Tanja Buttler (TUD), Collecting Lessons Learned
49 Gleb Polevoy (TUD), Participation and Interaction in Projects. A Game-Theoretic Analysis
50 Yan Wang (UVT), The Bridge of Dreams: Towards a Method for Operational Performance Alignment in IT-enabled Service Supply Chains

---

2017 01 Jan-Jaap Oerlemans (UL), Investigating Cybercrime
02 Sjoerd Timmer (UU), Designing and Understanding Forensic Bayesian Networks using Argumentation
03 Daniël Harold Telgen (UU), Grid Manufacturing; A Cyber-Physical Approach with Autonomous Products and Reconfigurable Manufacturing Machines
04 Mrunal Gawade (CWI), Multi-core Parallelism in a Column-store
05 Mahdieh Shadi (UVA), Collaboration Behavior
06 Damir Vandic (EUR), Intelligent Information Systems for Web Product Search
07 Roel Bertens (UU), Insight in Information: from Abstract to Anomaly
08 Rob Konijn (VU) , Detecting Interesting Differences:Data Mining in Health Insurance Data using Outlier Detection and Subgroup Discovery
09 Dong Nguyen (UT), Text as Social and Cultural Data: A Computational Perspective on Variation in Text
10 Robby van Delden (UT), (Steering) Interactive Play Behavior
11 Florian Kunneman (RUN), Modelling patterns of time and emotion in Twitter #anticipointment
12 Sander Leemans (TUE), Robust Process Mining with Guarantees
13 Gijs Huisman (UT), Social Touch Technology - Extending the reach of social touch through haptic technology
14 Shoshannah Tekofsky (UvT), You Are Who You Play You Are: Modelling Player Traits from Video Game Behavior
15 Peter Berck (RUN), Memory-Based Text Correction
16 Aleksandr Chuklin (UVA), Understanding and Modeling Users of Modern Search Engines
17 Daniel Dimov (UL), Crowdsourced Online Dispute Resolution
18 Ridho Reinanda (UVA), Entity Associations for Search
19 Jeroen Vuurens (UT), Proximity of Terms, Texts and Semantic Vectors in Information Retrieval
20 Mohammadbashir Sedighi (TUD), Fostering Engagement in Knowledge Sharing: The Role of Perceived Benefits, Costs and Visibility

21 Jeroen Linssen (UT), Meta Matters in Interactive Storytelling and Serious Gaming (A Play on Worlds)
22 Sara Magliacane (VU), Logics for causal inference under uncertainty
23 David Graus (UVA), Entities of Interest — Discovery in Digital Traces
24 Chang Wang (TUD), Use of Affordances for Efficient Robot Learning
25 Veruska Zamborlini (VU), Knowledge Representation for Clinical Guidelines, with applications to Multimorbidity Analysis and Literature Search
26 Merel Jung (UT), Socially intelligent robots that understand and respond to human touch
27 Michiel Joosse (UT), Investigating Positioning and Gaze Behaviors of Social Robots: People's Preferences, Perceptions and Behaviors
28 John Klein (VU), Architecture Practices for Complex Contexts
29 Adel Alhuraibi (UvT), From IT-BusinessStrategic Alignment to Performance: A Moderated Mediation Model of Social Innovation, and Enterprise Governance of IT"
30 Wilma Latuny (UvT), The Power of Facial Expressions
31 Ben Ruijl (UL), Advances in computational methods for QFT calculations
32 Thaer Samar (RUN), Access to and Retrievability of Content in Web Archives
33 Brigit van Loggem (OU), Towards a Design Rationale for Software Documentation: A Model of Computer-Mediated Activity
34 Maren Scheffel (OU), The Evaluation Framework for Learning Analytics
35 Martine de Vos (VU), Interpreting natural science spreadsheets
36 Yuanhao Guo (UL), Shape Analysis for Phenotype Characterisation from High-throughput Imaging
37 Alejandro Montes Garcia (TUE), WiBAF: A Within Browser Adaptation Framework that Enables Control over Privacy
38 Alex Kayal (TUD), Normative Social Applications
39 Sara Ahmadi (RUN), Exploiting properties of the human auditory system and compressive sensing methods to increase noise robustness in ASR
40 Altaf Hussain Abro (VUA), Steer your Mind: Computational Exploration of Human Control in Relation to Emotions, Desires and Social Support For applications in human-aware support systems
41 Adnan Manzoor (VUA), Minding a Healthy Lifestyle: An Exploration of Mental Processes and a Smart Environment to Provide Support for a Healthy Lifestyle
42 Elena Sokolova (RUN), Causal discovery from mixed and missing data with applications on ADHD datasets
43 Maaike de Boer (RUN), Semantic Mapping in Video Retrieval
44 Garm Lucassen (UU), Understanding User Stories - Computational Linguistics in Agile Requirements Engineering
45 Bas Testerink (UU), Decentralized Runtime Norm Enforcement
46 Jan Schneider (OU), Sensor-based Learning Support
47 Jie Yang (TUD), Crowd Knowledge Creation Acceleration
48 Angel Suarez (OU), Collaborative inquiry-based learning

2018 01 Han van der Aa (VUA), Comparing and Aligning Process Representations

02 Felix Mannhardt (TUE), Multi-perspective Process Mining
03 Steven Bosems (UT), Causal Models For Well-Being: Knowledge Modeling, Model-Driven Development of Context-Aware Applications, and Behavior Prediction
04 Jordan Janeiro (TUD), Flexible Coordination Support for Diagnosis Teams in Data-Centric Engineering Tasks
05 Hugo Huurdeman (UVA), Supporting the Complex Dynamics of the Information Seeking Process
06 Dan Ionita (UT), Model-Driven Information Security Risk Assessment of Socio-Technical Systems
07 Jieting Luo (UU), A formal account of opportunism in multi-agent systems
08 Rick Smetsers (RUN), Advances in Model Learning for Software Systems
09 Xu Xie (TUD), Data Assimilation in Discrete Event Simulations
10 Julienka Mollee (VUA), Moving forward: supporting physical activity behavior change through intelligent technology
11 Mahdi Sargolzaei (UVA), Enabling Framework for Service-oriented Collaborative Networks
12 Xixi Lu (TUE), Using behavioral context in process mining
13 Seyed Amin Tabatabaei (VUA), Computing a Sustainable Future
14 Bart Joosten (UVT), Detecting Social Signals with Spatiotemporal Gabor Filters
15 Naser Davarzani (UM), Biomarker discovery in heart failure
16 Jaebok Kim (UT), Automatic recognition of engagement and emotion in a group of children
17 Jianpeng Zhang (TUE), On Graph Sample Clustering
18 Henriette Nakad (UL), De Notaris en Private Rechtspraak
19 Minh Duc Pham (VUA), Emergent relational schemas for RDF
20 Manxia Liu (RUN), Time and Bayesian Networks
21 Aad Slootmaker (OUN), EMERGO: a generic platform for authoring and playing scenario-based serious games
22 Eric Fernandes de Mello Araujo (VUA), Contagious: Modeling the Spread of Behaviours, Perceptions and Emotions in Social Networks
23 Kim Schouten (EUR), Semantics-driven Aspect-Based Sentiment Analysis
24 Jered Vroon (UT), Responsive Social Positioning Behaviour for Semi-Autonomous Telepresence Robots
25 Riste Gligorov (VUA), Serious Games in Audio-Visual Collections
26 Roelof Anne Jelle de Vries (UT),Theory-Based and Tailor-Made: Motivational Messages for Behavior Change Technology
27 Maikel Leemans (TUE), Hierarchical Process Mining for Scalable Software Analysis
28 Christian Willemse (UT), Social Touch Technologies: How they feel and how they make you feel
29 Yu Gu (UVT), Emotion Recognition from Mandarin Speech
30 Wouter Beek, The "K" in "semantic web" stands for "knowledge": scaling semantics to the web

2019 01 Rob van Eijk (UL),Web privacy measurement in real-time bidding systems. A graph-based approach to RTB system classification
02 Emmanuelle Beauxis Aussalet (CWI, UU), Statistics and Visualizations for Assessing Class Size Uncertainty
03 Eduardo Gonzalez Lopez de Murillas (TUE), Process Mining on Databases: Extracting Event Data from Real Life Data Sources
04 Ridho Rahmadi (RUN), Finding stable causal structures from clinical data
05 Sebastiaan van Zelst (TUE), Process Mining with Streaming Data
06 Chris Dijkshoorn (VU), Nichesourcing for Improving Access to Linked Cultural Heritage Datasets
07 Soude Fazeli (TUD), Recommender Systems in Social Learning Platforms
08 Frits de Nijs (TUD), Resource-constrained Multi-agent Markov Decision Processes
09 Fahimeh Alizadeh Moghaddam (UVA), Self-adaptation for energy efficiency in software systems
10 Qing Chuan Ye (EUR), Multi-objective Optimization Methods for Allocation and Prediction
11 Yue Zhao (TUD), Learning Analytics Technology to Understand Learner Behavioral Engagement in MOOCs
12 Jacqueline Heinerman (VU), Better Together
13 Guanliang Chen (TUD), MOOC Analytics: Learner Modeling and Content Generation
14 Daniel Davis (TUD), Large-Scale Learning Analytics: Modeling Learner Behavior & Improving Learning Outcomes in Massive Open Online Courses
15 Erwin Walraven (TUD), Planning under Uncertainty in Constrained and Partially Observable Environments
16 Guangming Li (TUE), Process Mining based on Object-Centric Behavioral Constraint (OCBC) Models
17 Ali Hurriyetoglu (RUN),Extracting actionable information from microtexts
18 Gerard Wagenaar (UU), Artefacts in Agile Team Communication
19 Vincent Koeman (TUD), Tools for Developing Cognitive Agents
20 Chide Groenouwe (UU), Fostering technically augmented human collective intelligence
21 Cong Liu (TUE), Software Data Analytics: Architectural Model Discovery and Design Pattern Detection
22 Martin van den Berg (VU),Improving IT Decisions with Enterprise Architecture
23 Qin Liu (TUD), Intelligent Control Systems: Learning, Interpreting, Verification
24 Anca Dumitrache (VU), Truth in Disagreement - Crowdsourcing Labeled Data for Natural Language Processing
25 Emiel van Miltenburg (VU), Pragmatic factors in (automatic) image description
26 Prince Singh (UT), An Integration Platform for Synchromodal Transport
27 Alessandra Antonaci (OUN), The Gamification Design Process applied to (Massive) Open Online Courses

28 Esther Kuindersma (UL), Cleared for take-off: Game-based learning to prepare airline pilots for critical situations
29 Daniel Formolo (VU), Using virtual agents for simulation and training of social skills in safety-critical circumstances
30 Vahid Yazdanpanah (UT), Multiagent Industrial Symbiosis Systems
31 Milan Jelisavcic (VU), Alive and Kicking: Baby Steps in Robotics
32 Chiara Sironi (UM), Monte-Carlo Tree Search for Artificial General Intelligence in Games
33 Anil Yaman (TUE), Evolution of Biologically Inspired Learning in Artificial Neural Networks
34 Negar Ahmadi (TUE), EEG Microstate and Functional Brain Network Features for Classification of Epilepsy and PNES
35 Lisa Facey-Shaw (OUN), Gamification with digital badges in learning programming
36 Kevin Ackermans (OUN), Designing Video-Enhanced Rubrics to Master Complex Skills
37 Jian Fang (TUD), Database Acceleration on FPGAs
38 Akos Kadar (OUN), Learning visually grounded and multilingual representations

2020 01 Armon Toubman (UL), Calculated Moves: Generating Air Combat Behaviour
02 Marcos de Paula Bueno (UL), Unraveling Temporal Processes using Probabilistic Graphical Models
03 Mostafa Deghani (UvA), Learning with Imperfect Supervision for Language Understanding
04 Maarten van Gompel (RUN), Context as Linguistic Bridges
05 Yulong Pei (TUE), On local and global structure mining
06 Preethu Rose Anish (UT), Stimulation Architectural Thinking during Requirements Elicitation - An Approach and Tool Support
07 Wim van der Vegt (OUN), Towards a software architecture for reusable game components