Exploratory search over semi-structured documents

Azarbonyad, H.

Exploratory search is an information seeking task in which users are not familiar with the specification of their information need and how to answer it and the system helps them in this regard. In this thesis, we study how metadata and structure associated with textual documents can be helpful in supporting exploratory search tasks. We first study how metadata and structure can be exploited to manage documents and support access to semi-structured documents more effectively. We define semi-structured documents as documents that beside a textual content, contain tags and other markers that separate documents into different parts such as title, body, metadata, etc. Then we study, in two case studies, how metadata and structure can help to support exploratory search tasks.

Exploratory Search over Semi-structured Documents

Hosein Azarbonyad

Exploratory Search over Semi-structured Documents

Hosein Azarbonyad

Exploratory Search over Semi-structured Documents

Hosein Azarbonyad

Exploratory Search over Semi-structured Documents

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het College voor Promoties ingestelde
commissie, in het openbaar te verdedigen in
de Agnietenkapel
op vrijdag 15 maart 2019, te 10:00 uur

door

Hosein Azarbonyad

geboren te Ardebil

Promotiecommissie

Promotoren:		
	Dr. ir. J. Kamps	Universiteit van Amsterdam
	Prof. dr. M. de Rijke	Universiteit van Amsterdam
Co-promotor:		
	Dr. M. Marx	Universiteit van Amsterdam
Overige leden:		
	Prof. dr. A. Betti	Universiteit van Amsterdam
	Prof. dr. E. Kanoulas	Universiteit van Amsterdam
	Prof. dr. ir. A. de Vries	Radboud Universiteit
	Dr. R. White	Microsoft Research Seattle
	Prof. dr. M. Worring	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

This research was supported by the Netherlands Organization for Scientific Research (ExPoSe project, NWO CI # 314.99.108; DiLiPaD project, NWO Digging into Data # 600.006.014), Nederlab (340-6148-t1-6), and by the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement ENVRI, number 283465.

Cover by Mostafa Dehghani
Printed by Off Page, Amsterdam, The Netherlands

ISBN: 978-94-6182-939-9

Acknowledgments

Doing a PhD has been an amazing journey for me. Looking back at the past four years, I can realize how much I grew as a person both professionally and personally over these years. This journey would not have been possible without the help, support, and guidance of my supervisors, family, and friends.

First of all, I would like to thank my main supervisor Maarten Marx, who trusted me and gave me the opportunity to do my PhD at ILPS. Maarten, you always encouraged me and I learned a lot from you. Thanks for being very patient, motivating, and supportive. I have a tendency to loose focus when doing research by getting overwhelmed with very small details, but you helped me very much with this and without this help I probably could not finish my research. We had a lot of productive discussions and I enjoyed them a lot. Beside work, you helped me a lot with many other things such as getting familiar with Dutch culture and adapting myself with it. I really appreciate it.

I also thank my promoter Maarten de Rijke for his support during my PhD. Maarten, you created such a fantastic group which I believe is the best place to do a PhD at. You are a big leader and the amount of support I got from you during my PhD was fascinating. Thanks for your help, your insightful guidance, and creating such a great atmosphere at ILPS.

I was very lucky to also have Jaap Kamps as my co-promoter. Jaap, you had a very big influence on my research when I was taking my initial steps to do a PhD. I had my first interviews with you and you encouraged and motivated me a lot to do my PhD at the UvA. Thank you for all your help during my PhD.

I am honoured to have Arianna, Evangelos, Arjen, Ryen, and Marcel as my committee members. Thanks a lot for agreeing to be in my PhD committee.

Many thanks to Keyvan and Mostafa for being my paranymphs and standing by my side in my defense. Keyvan, thank you for being my friend, home-mate, and paranymph. We shared every single experience in our PhD with each other. Thank you so much for walking along in the PhD path with me. I am very lucky to have such a wise and cheerful friend. Dear Mostafa, we came together to the Netherlands to start our PhDs. I cannot imagine how my PhD life would go without having you as a friend and colleague. We did most of our work together and I enjoyed every single moment of our collaboration. I am very happy we ended up doing our PhDs together at ILPS.

I would like to thank Mostafa for designing the cover of this thesis and Rolf for translating thesis summary to Dutch.

I was a great pleasure and honour for me to work with very talented and interesting people at ILPS. Beyond work, I enjoyed spending time with you over drinks, our social events, running sessions, playing squash, and volleyball. Our Friday afternoon drinks created many good memories for me. I enjoyed every second of being at ILPS mostly because for your nice company. Thanks a lot: Adith, Aldo, Aleksandr, Alexey, Amir, Ana, Anna, Anne, Arianna, Artem, Bob, Boris, Chang, Christof, Christophe, Chuan, Cristina, Daan, Damien, Dan, Dat, David, David, Dilek, Eva, Evangelos, Evgeny, Fei, Hamid, Harrie, Hendra, Hendrik, Hendrike, Hinda, Isaac, Ilya, Ivan, Jiahuan, Jie, Julia, Julien, Kaspar, Katya, Ke, Lars, Maarten, Maarten, Maartje, Manos, Marc, Marlies, Marzieh, Masrour, Maurits, Mostafa, Nikos, Pengjie, Petra, Praveen, Richard, Ridho, Rolf, Shangsong, Shaojie, Spyretta, Svitlana, Thorsten, Tobias, Tom, Trond, Vera, Xiaohui, Xiaojuan, Xinyi, Yaser, Yangjun, Yifan, Zhaochun and Ziming. I would like to especially thank Ana, Bob, Chang, Christophe, Harrie, Julia, Nikos, Rolf, and Ziming for all the nice conversations we had and all the fun stuff we did together.

I had a chance to do an industrial internship at Microsoft Research which was a very valuable experience for me. I would like to thank Ryen and Rob and the whole Cortana Research team for their help and guidance during this internship.

I enjoy living in Amsterdam and one of the reasons is the wonderful friends I have here who indirectly had an influence in my PhD. I want to especially thank Keyvan, Mojtaba, Mehran, Hadi, Reza, Mostafa, Samira, and Masoud. Mojtaba, thanks for always being cheerful and taking the initiative to bring fun to our lives. You are a wonderful person with a big heart. Mehran, you cannot imagine how great it is to have an amazing friend from my hometown here in Amsterdam. Thank you for being here and for being one of my best friends. Hadi, thank you for all adventures we had and all good memories we made together (how silly it was to get lost in a mountain in Chile). And Shaghayegh, thank you for all your love and support. You are amazing and I am very happy I met you. We have a very wonderful Iranian community in the Netherlands. I always enjoyed spending time with you. Thank you: Ali, Amir, Amir Hossein, Amin, Babak, Dena, Erfan, Fahimeh, Fereshteh, Hadi, Hamed, Hamid, Hamraz, Hoda, Hodjat, Iman, Irene, Keyvan, Leila, Maryam, Marzieh, Marzieh, Masoud, Mehran, Mina, Mojdeh, Mojtaba, Mohammad, Mohammad Reza, Mostafa, Nasrin, Niloofar, Parisa, Reza, Samira, Sara, Sima, Tannaz, and Yaser.

Last but not least, I express my deepest gratitude to my parents and to my siblings: Ahmad, Safoora, Adele, Mohammad, Ali, Roghayye. You are always my main driving motivation. Nothing is better than knowing you have unconditional love and support in your life. Thank you for giving it me.

Contents

1 Introduction

Digital documents on the Web are a heterogeneous mixture of text, metadata and structure [108, 113]. The structure and metadata associated with documents can be helpful in many tasks including search and managing documents [44, 100, 173, 208]. For example, tags associated with documents can be used to organize documents and support faceted and exploratory search over them [173]. If the metadata associated with documents comes from a thesaurus, we can automatically organize documents into a hierarchical structure [71]. Temporal metadata such as creation date can be used to organize documents from a different angle and help users to track topics over time [33].

In this thesis, we study how metadata and structure associated with textual documents can be helpful in supporting exploratory search tasks. We first study how metadata and structure can be exploited to manage documents and support access to semi-structured documents more effectively. We define semi-structured documents as documents that beside a textual content, contain tags and other markers that separate documents into different parts such as title, body, metadata, etc. Then we study, in two case studies, how metadata and structure can improve exploratory search tasks. Exploratory search is a very broad concept and a system should have several features to support exploratory search activities [132, 205, 206]. White and Roth [205] characterize different aspects of exploratory search systems. We restrict our attention to some specific aspect of exploratory search including "offering facets and metadata-based result filtering" as we use metadata features during the two case studies and "supporting querying and rapid query refinement" as the task in the case studies support this functionality.

To support exploratory search, we first focus on document management tools that can make access to the data easier. In this thesis, document management is defined as the task of grouping similar documents together and classifying them. We define similarity between documents from different angles. The first angle is topical similarity in which our goal is to use content, metadata, and structure in a unified way to classify documents based on how similar their topics are. We consider a scenario in which there is a hierarchical thesaurus with a high number of entries and the challenge is to correctly assign documents to possibly multiple entries in the thesaurus. Next, we take a different angle and measure topical diversity of documents and use it for managing documents. We are interested in topical diversity as it has been shown that topical diversity is an important characteristic of interesting documents [15, 61]. We determine the main drawbacks with existing models for measuring diversity and propose an approach that addresses them. As the third angle, we move to the domain of managing emails and focus on how we can help users manage their tasks created via email efficiently. Nowadays, email is used as a tool to create and manage tasks [20, 43, 145, 189]. When the number of ongoing tasks created via email increases, people can struggle to manage their tasks and monitor their progress [17, 207]. We focus on commitments made by people via email as one type of task and study how we can build an automatic commitment detection method.

We then consider two different exploratory search tasks and study how metadata and structure can be helpful in these tasks. The first task is measuring semantic shifts in the meaning of words. We propose an approach for detecting semantic shifts between different viewpoints—broadly defined as a set of texts that share a specific metadata feature, which can be a time-period, but also a social entity such as a political party. We study how detected shifts can be helpful in various tasks including monitoring changes in meanings, summarizing diverging viewpoints, and ideology detection in a political setting. The second case study lies in the domain of community question answering as it happens in fora such as Stack Exchange.[1] We study the task of finding existing questions (often with an answer already) that are similar to a given question.

1.1 Research Outline and Questions

This thesis contains two research themes: using content, structure, and metadata (1) to support access to semi-structured documents (Chapters 2, 3 and 4) and (2) exploratory search over semi-structured documents (Chapters 5 and 6). Below, we list the main research question of every chapter.

1.1.1 Managing semi-structured documents

For our first research theme, we study how structure and metadata can be used for classifying documents based on their topics (Chapter 2) and how interesting they are (Chapter 3). We then focus on the email management domain and study how we can help users manage their emails (Chapter 4).

We first study how we can exploit structure and metadata to classify documents topically. Multi-Label Text Classification (MLTC) is a supervised machine learning task in which the goal is to learn a classifier that assigns multiple labels to text documents [79]. Efficiently exploiting all sources of information associated with semi-structured documents such as labels, their structure, the representation of the labels and relations between them has a high impact on the performance of MLTC systems. Most of the current approaches to MLTC only use labeled documents as the primary source of information for MLTC. We investigate the effectiveness of different sources of information such as the labeled training data, textual labels of classes, and taxonomy relations of classes for MLTC. In doing this, we answer the following research question:

RQ1 How helpful is integrating a variety of sources of information extracted from

[1] `https://stackexchange.com`

content, structure, and metadata as features to improve the performance of MLTC systems?

To answer this research question, first, for each document-class pair, different features are extracted using different sources of information. The features reflect the similarity of classes and documents. Then, MLTC is considered to be a ranking problem and learning to rank (LTR) is used to rank class-labels given a document. To further improve performance, we apply score propagation on top of LTR based on the co-occurrence patterns of classes in labeled documents.

Next, we focus on the task of managing documents based on their topical diversity. A high degree of topical diversity has been considered to be an important characteristic of interesting text documents [15, 61]. A recent proposal for measuring topical diversity identifies three distributions for assessing the diversity of documents: distributions of words within documents, words within topics, and topics within documents [15]. Topic models play a central role in this approach and, hence, their quality is crucial to the efficacy of measuring topical diversity. The quality of topic models is affected by two dimensions: generality and impurity of topics. General topics only include common information of a background corpus and are assigned to most of the documents, and so overestimate their diversity. Impure topics contain words that are not related to the topic. Impurity lowers the interpretability of topic models and impure topics are likely to get assigned to documents erroneously. We propose a hierarchical re-estimation process aimed at removing generality and impurity, and at answering the following research question:

RQ2 How effective is our hierarchical re-estimation approach in measuring topical diversity of documents? Are the topic models that have been improved in this way also successfully applicable for other tasks such as documents clustering and classification?

To answer this research question, we focus on different components of topic models and re-estimate them at three different levels: (1) document re-estimation, which removes general words from the documents; (2) topic re-estimation, which re-estimates the distribution over words of each topic; and (3) topic assignment re-estimation, which re-estimates for each document its distributions over topics. We show that for measuring topical diversity of text documents, our re-estimation approach improves over the state-of-the-art.

In the last chapter of this part, we study how we can help users manage their emails more effectively. People use email not only as a communication tool, but also as a means to create and manage tasks [20, 43, 145, 189]. When the number of ongoing tasks created via email increases, people can struggle to manage their tasks and monitor their progress [17, 207]. People often make commitments to perform future actions. Detecting commitments made in email (e.g., “I will send the draft of my thesis by end of this week.”) enables digital assistants to help their users recall promises they have made and made to them. Despite the potential benefits of automatic commitment detection, work in this important area has been limited to only a handful of studies [41, 50, 115, 116, 128]. We focus on this task and answer the following research question:

RQ3 Can commitments be reliably detected in emails? And how does the performance of commitment models change when they are tested on a different domain than they are trained on?

To answer **RQ3**, we build the first large-enough dataset for the task of commitment detection. Using this dataset, we show that commitments can be reliably extracted from emails when models are trained and evaluated on the same domain (corpus). However, their performance degrades when the evaluation domain differs. This illustrates the domain bias associated with email datasets and a need for more robust and generalizable models for commitment detection. We characterize the differences between domains (email corpora) and use this characterization to transfer knowledge between them and create reliable domain-independent commitment models. Our findings illustrate that domain differences can have a significant negative impact on the quality of commitment detection models and that transfer learning has potential to address this issue.

1.1.2 Using structure and metadata for exploratory search

We study how structure and metadata are helpful in supporting exploratory search tasks such as tracking semantic shifts (Chapter 5) and finding similar questions (Chapter 6).

We first focus on the use of metadata features to detect shifts in the meaning of words and track them. Detecting and characterizing such shifts can be helpful for search and exploration of historical collections. Due to its importance, recently, researchers started to pay attention to the detection of temporal shifts in the meaning of words [75, 90, 104, 110]. Most (if not all) of these approaches are restricted to changes over time. We focus on detecting semantic shifts between different viewpoints–broadly defined as a set of texts that share a specific metadata feature, which can be a time-period, but also a social entity such as a political party. We answer the following research question:

RQ4 How can we use metadata information to measure semantic shifts? How effective are the proposed approaches in quantifying the changes in word meaning over various dimensions such as time and political context?

To answer this research question, for each viewpoint, we learn a semantic space in which each word is represented as a low dimensional neural embedded vector. The challenge is to compare the meaning of a word in one space to its meaning in another space and measure the size of the semantic shifts. We compare the effectiveness of a measure based on optimal transformations between the two spaces with a measure based on the similarity of the neighbors of the word in the respective spaces. We find that semantic shifts not only occur over time, but also along different viewpoints in a short period of time. Moreover, we show that the information about semantic shifts contributes to other tasks such as summarization and classification.

In the last chapter, we move to the task of question retrieval in Community Question Answering (cQA) forums. cQA forums are widely used sources of knowledge [46, 142, 225, 231]. Similar question retrieval is an important task in cQA that can help address issues such as question starvation and duplicate question detection [46, 142, 152]. Two challenges with question retrieval in cQA are (1) the vocabulary gap, which is the phenomenon that users express the same question with different words, and (2) expressive inefficiency, which is the phenomenon that users do not explain their question with enough details. It has been shown that semantic matching of questions can address the vocabulary gap to some extent, however, there are not many effective methods to address both challenges [46, 152, 225, 231]. We propose a method that not only estimates semantic similarity of questions but also exploits the structure and different fields of questions to further bridge the vocabulary gap and address the expressive inefficiency problem. We answer the following research question:

RQ5 Does using the rich data and structure available on question answering forums lead to a better score than the state of the art on the task of question retrieval?

We propose a multi-context neural attention mechanism to weight different sources of information associated with questions to determine semantic similarity. We show that the attention mechanism is a powerful means to exploit different sources of information for the question retrieval task. Our findings indicate that for semantic matching of questions it is important to efficiently exploit all sources of information and augment semantic matching models with the rich content and structure associated with questions on typical cQA platforms.

1.2 Main Contributions

In this section, we list theoretical, algorithmic and empirical contribution of the thesis. For each contribution, we list the chapter from which it originates.

1.2.1 Theoretical contributions

1. Introducing the concept of word stability and a general framework for computing semantic shifts by using word embeddings trained on corpora that (are presumed to) represent specific viewpoints. (Chapter 5)

1.2.2 Algorithmic contributions

2. An LTR and score propagation approach for classifying documents for the MLTC task. (Chapter 2)

3. A hierarchical model for removing generality and impurity from topic models and measuring topical diversity. (Chapter 3)

4. A neural model for characterizing domain bias in email datasets and removing it from commitment models. (Chapter 4)

5. Several methods to measure the amount of change in the meaning of words over various dimensions such as time and social groups. (Chapter 5)

6. A recurrent neural attentive model for estimating similarity of questions using multiple fields in cQA forums. (Chapter 6)

1.2.3 Empirical contributions

7. (a) Analysis of the effectiveness of different sources of information for MLTC and advice on how to create a lean-and-mean effective and efficient classifier. (b) An empirical comparison of classification and ranking-based approaches for MLTC. (c) An empirical comparison of static and dynamic approaches for selecting the number of classes for documents. (Chapter 2)

8. (a) Evaluation of different topic models on different tasks including topical diversity, document clustering, and document classification. (b) Analysis on the effect of impurity and generality on the quality of topic models. (Chapter 3)

9. (a) A dataset for detecting commitments and evaluating the performance of different commitment models on it. (b) A study on the impact of domain transfer on commitment detection and showing that the quality of commitment detection degrades significantly as we apply commitment detection models across domains. (c) A characterization of differences between email corpora and showing that domain adaptation (specifically, transfer learning) can remove domain-specific bias from commitment detection models. (Chapter 4)

10. (a) An analysis to show that semantic shifts not only occur over time, but also across different viewpoints in a short period of time. (b) An evaluation dataset for detecting semantic shifts and contrastive viewpoint summarization. (c) An extensive analysis of word stability measure in different tasks including detecting shifts, summarization, and classification. (Chapter 5)

11. (a) A comparison of the performance of different question retrieval methods and their effectiveness in addressing the issues associated with this task. (b) Extensive analyses to assess the importance of different fields of questions for learning representations of questions. (c) Analysis of the effectiveness of a multi-context attention mechanism for exploiting multiple fields associated with questions. (Chapter 6)

1.3 Thesis Overview

This thesis is organized in two parts: *managing semi-structured documents* and *using structure and metadata for exploratory search*.

The first part consists of three chapters. In Chapter 2 we study how metadata and structure can be used to automatically assign labels from a thesaurus to documents using supervised learning; in Chapter 3 we apply unsupervised learning to assign topics to documents; and in Chapter 4 we move to the email management domain and study how we can help users by identifying commitments in emails.

In the second part, we study how metadata and structure of documents can be used beside their content to support two different exploratory search tasks. In Chapter 5, we use metadata such as time stamps and social groups tags associated with documents to measure and track semantic shifts. Chapter 6 is centered around the task of question retrieval in cQA forums and how metadata and structure can address challenges associated with this task.

Finally, in Chapter 7, we conclude the thesis and discuss limitations and future directions.

There is no particular dependence between the chapters of the thesis and they can be read independently.

1.4 Origins

In this section, we list the publications each chapter is based on and explain the role of each author.

- **Chapter 2** is based on the following papers:

  - H. Azarbonyad, M. Dehghani, M. Marx, and J. Kamps. Learning to rank for multi label text classification: Combining different sources of information. *Journal of Natural Language Engineering, under review*, 2018 [12]
  - M. Dehghani, H. Azarbonyad, M. Marx, and J. Kamps. Sources of evidence for automatic indexing of political texts. In *Proceedings of the 37th European Conference on IR Research*, ECIR '15, pages 568–573, 2015 [51]

  HA designed the algorithm, ran the experiments, and did most of the writing; MD helped with the algorithm design and running the experiments; MM, and JK contributed to the writing.

- **Chapter 3** is based on the following paper:

  - H. Azarbonyad, M. Dehghani, T. Kenter, M. Marx, J. Kamps, and M. de Rijke. HiTR: Hierarchical topic model re-estimation for measuring topical diversity of documents. *IEEE Transactions on Knowledge and Data Engineering, to appear*, 2018 [10]
  - H. Azarbonyad, M. Dehghani, T. Kenter, M. Marx, J. Kamps, and M. de Rijke. Hierarchical re-estimation of topic models for measuring topical diversity. In *Proceedings of the 39th European Conference on IR Research*, ECIR '17, pages 68–81, 2017 [9]
  - H. Azarbonyad, F. Saan, M. Dehghani, M. Marx, and J. Kamps. Are topically diverse documents also interesting? In *Proceedings of the 6th International Conference on Experimental IR Meets Multilinguality, Multimodality, and Interaction*, CLEF '15, pages 215–221, 2015 [7]

  HA designed the algorithm, ran the experiments, and did most of the writing; MD helped with the algorithm design; TK, MM, JK, MdR contributed to the writing.

- **Chapter 4** is based on the following paper:

  - H. Azarbonyad, R. Sim, and R. W.White. Domain adaptation for commitment detection in email. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, 2018 [13]

This work was done during an internship in Microsoft Cortana Research in 2018. The task was proposed by RS and RW; HA designed the algorithms, ran experiments, and did most of the writing; RS further helped with running experiments. RS and RW contributed to the writing.

- **Chapter 5** is based on the following paper:

  - H. Azarbonyad, M. Dehghani, K. Beelen, A. Arkut, M. Marx, and J. Kamps. Words are malleable: Computing semantic shifts in political and media discourse. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 1509–1518, 2017 [8]

  HA designed the algorithm, ran the experiments, and did most of the writing; AA and KB helped with building datasets; AA further helped with algorithm design; MD, MM, JK contributed to the writing.

- **Chapter 6** is based on the following paper:

  - H. Azarbonyad, M. Dehghani, M. Marx, and M. de Rijke. Learning question representations for question retrieval using content, structure and attention. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, under review, 2018 [11]

  HA designed the algorithm, ran the experiments, and did most of the writing; MD helped with the algorithm design; MM, MD, MdR contributed to the writing.

The thesis also indirectly builds on the following papers (listed in reverse chronological order):

- M. Dehghani, H. Azarbonyad, J. Kamps, and M. de Rijke. Learning to transform, combine, and reason in open-domain question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, 2018 [60]

- M. Dehghani, G. Jagfeld, H. Azarbonyad, A. Olieman, J. Kamps, and M. Marx. On search powered navigation. In *Proceedings of the 2017 ACM International Conference on the Theory of Information Retrieval*, ICTIR '17, pages 317–320, 2017 [58]

- M. Dehghani, H. Azarbonyad, J. Kamps, and M. de Rijke. Share your model instead of your data: Privacy preserving mimic learning for ranking. In *SIGIR wokshop on Neural Information Retrieval*, 2017 [57]

- M. Dehghani, G. Jagfeld, H. Azarbonyad, A. Olieman, J. Kamps, and M. Marx. Telling how to narrow it down: Browsing path recommendation for exploratory search. In *Proceedings of the ACM SIGIR Conference on Human Information Interaction and Retrieval*, CHIIR '17, pages 369–372, 2017 [59]

- M. Dehghani, H. Azarbonyad, J. Kamps, D. Hiemstra, and M. Marx. Luhn revisited: Significant words language models. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '16, pages 1301–1310, 2016 [52]

- H. Azarbonyad and E. Kanoulas. Power analysis for interleaving experiments by means of offline evaluation. In *Proceedings of the 2017 ACM International Conference on the Theory of Information Retrieval*, ICTIR '17, pages 87–90, 2016 [5]

- M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. On horizontal and vertical separation in hierarchical text classification. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ICTIR '16, pages 185–194, 2016 [55]

- M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. Two-way parsimonious classification models for evolving hierarchies. In *Proceedings of the 7th International Conference on Experimental IR Meets Multilinguality, Multimodality, and Interaction*, CLEF '16, pages 69–82, 2016 [53]

- H. Azarbonyad. Measuring interestingness of political documents. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 1175–1175, 2016 [4]

- M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. Generalized group profiling for content customization. In *Proceedings of the ACM SIGIR Conference on Human Information Interaction and Retrieval*, CHIIR '16, pages 245–248, 2016 [54]

- M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. Significant words language models for contextual suggestion. In *Proceedings National Institute for Standards and Technology*, 2016 [56]

- H. Azarbonyad, M. Dehghani, M. Marx, and J. Kamps. Time-aware authorship attribution for short text streams. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 727–730, 2015 [6]

- A. Olieman, H. Azarbonyad, M. Dehghani, J. Kamps, and M. Marx. Entity linking by focusing dbpedia candidate entities. In *SIGIR workshop on Entity Recognition and Disambiguation*, pages 13–24, 2014 [148]

Part I

Managing Semi-structured Documents

2

Learning to Rank for Multi Label Text Classification: Combining Different Sources of Information

In this chapter, we address **RQ1** and focus on using content, structure, and metadata for classifying documents in the Multi-Label Text Classification (MLTC) task.

2.1 Introduction

MLTC is a supervised machine learning task in which the goal is to learn a classifier that assigns multiple labels to text documents [79]. MLTC has many applications in the real world, e.g., when a text document is about both politics and economics and we want to label it. Simple classification approaches become computationally expensive when the number of classes is high as for each class a different classifier should be trained [22]. To achieve a good performance in MLTC, instead of optimizing a model for each class separately, the model should be optimized with respect to a global optimum considering all classes. Learning to rank (LTR) has been shown to be an effective approach for MLTC [219]. In this approach, a model is trained to rank classes regarding the documents and select the *top k* classes as labels of documents. Rather than creating and optimizing a separate model for each class and predicting the probability of assigning each class to the given document, the learning objective of the LTR approach for MLTC is to create a global ranking model that ranks all classes for a given document.

In this chapter, we integrate a variety of sources of available information for MLTC using an LTR approach. There are different sources of information for the selection of appropriate classes for documents in the MLTC task, such as several document and class representations. The main aim of our approach is to effectively combine these sources of information in a supervised process. For example, classes can be expanded by their textual labels, which is useful for calculating the similarity of a class with the content of documents [171]. Moreover, the relations between classes (structure of thesauri), if existing, could be another useful source for finding semantic relations

This chapter was published as [12].

between classes and taking these relations into account [166, 171]. One of the main sources of information is a set of annotated documents with the labels assigned. These documents are used by supervised approaches as training data to create a classification model [143, 155]. In this study, we utilize various information sources for finding similarities of documents and classes and rank classes based on these similarity scores. Using each source of information, we define different features, each reflecting the similarity of documents and classes in a different way. We use an LTR approach for combining different sources of information. Similar to [219], we consider each document to be annotated as a query, and use all information associated with a class as a document. We choose LTR as the combination method since it is an effective approach for combining different types of signals. Moreover, it gives us the ability to analyze the contribution of different sources of information in the classification task. The proposed approach can be applied on any document collection that contains the mentioned sources of information.

In MLTC, there are implicit and explicit relations between classes that can be useful for classifying documents more accurately. We model the implicit relations between classes based on their co-occurrence patterns in the labeled documents and study how these patterns can help classify the documents more accurately. To do this, we propose a score propagation approach that re-estimates the similarity of classes and documents based on co-occurrence patterns of classes. For a class c, its similarity to a document d is smoothed (by linear interpolation) with another score that is based on the probability of c co-occurring with other classes c' and the similarity of c' to d.
In this chapter, we focus on addressing **RQ1**:

> *How helpful is integrating a variety of sources of information extracted from content, structure, and metadata as features to improve the performance of MLTC systems?*

We break down this research question to two research questions:
RQ1.1: How effective is a LTR approach integrating a variety of sources of information as features for MLTC?

To answer this research question, we evaluate the performance of the proposed LTR approach on the English version of JRC-Acquis [179] and compare our results with JEX [180], which is one of the state of the art systems developed for classifying JRC-Acquis documents. The results show that the LTR approach improves JEX by 20% in terms of Precision@5.
RQ1.2: Is it worthwhile to use the co-occurrence patterns of classes for MLTC?

To answer our second research question, we analyze the effect of score propagation on the performance of different approaches for MLTC and compare the results achieved by the propagated and the non-propagated versions of each method. The results indicate that propagated versions of all methods outperform their non-propagated counterparts.

Our main contributions are the following:

- We propose a framework for exploiting multiple sources of information including labeled documents, taxonomy relations of classes, and textual labels of classes for MLTC. We define different features using these sources of information and instead of training a classifier per class, create a ranking model that can rank classes based on their similarities to documents.

- We propose a score propagation approach to consider the co-occurrence patterns of classes in the labeled documents. The proposed approach can be applied on top of any classifier.

- We make the designed tool publicly available.[1] It automatically pre-processes textual data, constructs different representations for documents and classes, and computes different similarity metrics for documents and classes. These similarity metrics are used to build feature vectors for document-class pairs. The feature vectors extracted using JRC-aquis dataset and Eurovoc concepts are in the LTR format that can be directly used by LTR algorithms to train ranking models.

2.2 Related Work

Our method for multi-label text classification touches on research in multiple areas. We review work in three directions: multi-label text classification, learning to rank for multi-label text classification, and automatic indexing of political documents.

2.2.1 Multi-label text classification

Multi-Label Text Classification (MLTC) is the task of classifying text documents to multiple classes [188]. A well-known approach for MLTC is training a different classifier for each class and ranking and selecting the classes with regards to the probability of documents belonging to them [227]. Using this approach, Qiu et al. [158] exploit the well-known SVM classifier for training a binary classification model for each category. Then, they use these models to determine the classes that a document belongs to. Moreover, they use the hierarchical structure of classes and try to maximize the margin between leafs of the hierarchy. Qin and Wang [156] study the effectiveness of MLTC based on SVM classifiers. In other studies, SVM classifiers are combined with other classifiers to improve their performance [221]. Instead of SVM, Nam et al. [143] and Jiang et al. [95] employ a neural network based model to determine the labels of documents. Vilar et al. [194] use a similar approach and estimate the multinomial distribution of documents across the classes. Tree-based classifiers are also adapted and used for MLTC [84]. Huang et al. [86] study the effectiveness of learning class-specific features for MLTC.

Simple classification approaches become computationally expensive and infeasible when the number of classes is high [22]. Babbar and Schölkopf [14] propose a scalable approach for multi-label classification by training one-versus-rest classifiers. Although their proposed approach is computationally efficient, the trained model is large. We consider MLTC a ranking problem and instead of using classification methods we use an LTR approach to rank classes given a document. We create only one small model which can be used to assign classes to documents.

The main characteristic of MLTC discriminating it from single-label text classification is that documents can have more than one label. Therefore, capturing the dependencies between classes and incorporating them in the classification process could

[1] The source code is available here: `https://github.com/HoseinAzarbonyad/MLC`

be useful to improve the accuracy of classifiers [21, 72, 77, 143, 163]. Ghamrawi and McCallum [72] capture these dependencies using Conditional Random Fields (CRF), obtaining better classification scores. Read et al. [163] chain the classifiers to use the dependency information of classes in the labeling process. Nam et al. [143] try to capture these dependencies using neural networks specialized for document classification. In this study, we use the implicit dependencies of classes in a score propagation framework. Unlike previous work, our approach for incorporating class dependencies is independent of the underlying method and can be applied on top of any classification method. It is noteworthy that a similar label propagation has been used for single-label text classification when the size of training set is small [170, 197].

To classify documents in MLTC, there is a need to first determine the number of classes to be assigned to documents. In the single-label classification task, only one class is assigned to each document. This is usually done by setting a threshold on the scores estimated for the documents and assigning documents with a higher score than the threshold to the positive and rest of the documents to the negative class [219]. This strategy does not work for MLTC, especially in the case of a ranking-based MLTC task, as in this task we have a ranked list of classes and a document can have more than one class. A common approach for choosing the number of classes in MLTC is calibrating the scores generated for each class, setting a threshold on the scores, and assigning classes with a higher score than the threshold to documents [89, 227]. A static approach (fixing a threshold and using it for all documents) or dynamic approach (learning from training samples and having different threshold values for different documents) can be used for setting the threshold. Some popular choices for fixed thresholds are zero, e.g., for SVM-like classifiers, and 0.5 for probabilistic classifiers such as logistic regression [29, 39, 163]. Another common static approach is setting a threshold on the number of classes directly instead of setting the threshold on the scores [180]. This is an effective approach when the variance of the number of classes for documents is low. The dynamic threshold is set using a training set in which samples are a set of pairs of ranked lists with scores and, for each ranked list, an optimal threshold that minimizes a classification loss such as false positives or false negatives given the ranked list [62, 159, 219, 226]. For samples in the training set, the optimal threshold can be determined, however, for test samples the threshold should be estimated. This is done by learning a mapping function based on training samples that takes a ranked list and maps it to a threshold. This strategy has been shown to be very effective for MLTC [62, 219]. Similarly, instead of learning a mapping from ranked lists of document to a threshold, the mapping can be learned to map ranked lists to the number of classes directly [185]. Besides these generic approaches, some ad-hoc thresholding strategies are also used in previous studies. These strategies are specific to the learning algorithms [66, 203] and cannot be applied on top of other methods.

2.2.2 Learning to rank for multi-label classification

LTR was proposed in the context of ad-hoc information retrieval in which the goal is to create a ranking model that ranks *documents* with respect to *queries*. The LTR approach have been used for constructing a ranking model to rank classes with respect to a given document and select the most probable classes for the document as its labels

[65, 99, 219]. Yang and Gopal [219] map MLTC to the ad-hoc retrieval problem and use LTR for learning a ranking model. When we view MLTC as a problem of ranking class labels given a document, we can use LTR to estimate a classifier: we simply rank all classes given a document and assign the top k classes (k to be determined by another classifier) as labels to the input document. We now briefly describe LTR in terms of our classification task. We assume that we can compute several features which indicate, given a document d and a class c, how much discriminatory information the feature provides to determine if c is a label of d or not. In LTR these measures are called *features*. The goal is to find an optimal linear combination of these features. Formally, given n features f_i, we are searching for weights $w_1, \ldots, w_n$ such that the function $f(c,d)$ defined in (2.1) optimally scores and ranks classes with regards to documents on some test set.

$$f(c,d) = w_1 \cdot f_1(c,d) + \ldots + w_n \cdot f_n(c,d). \tag{2.1}$$

Similar to [219], we use LTR for MLTC. Using LTR for MLTC has many advantages compared to using traditional approaches for MLTC such as SVM's. Yang and Gopal [219] show that LTR outperforms classification-based approaches for MLTC by a large margin on a wide variety of datasets with different types of samples, e.g audio, image, and text. They used meta-level features for building an LTR model. The meta-level features are defined based on the distance between classes and documents. Fauzan and Khodra [65] use the same framework for classifying documents, however, they focus on text classification and instead of using meta-level features, used typical features such as TF-IDF weights of words for learning a LTR model. Their method also outperforms traditional classification-based approaches. Ju et al. [99] try to extend this idea by modeling the hierarchical structure of labels in the LTR framework. They use LTR as a re-ranker to re-rank the rankings created by a classifier by incorporating the structure of the categories. They achieve similar results to [219] confirming the effectiveness of LTR for modeling the hierarchical structure of labels for MLTC.

LTR can learn a global ranking function with respect to all classes, while classification-based approaches try to optimize a classifier locally for each class. In this sense, LTR can also take the relations between classes into account to some extent, which is a core problem in MLTC [219]. From an efficiency point of view, the constructed model of LTR is much smaller than the models constructed by traditional classifiers. Moreover, when using LTR, during inference only one model is used to score instances. Traditional classifiers build a model for each class and use all of them during inference which is less efficient compared to the LTR approach.

2.2.3 Automatic classification of political documents

Supervised classification of political text including parliamentary proceedings, legislative text, and news articles is an active research area [9, 53, 55, 179, 193]. While standard classification approaches such as SVM-based models are used for classification of news articles and parliamentary proceedings, specialized classification tools are developed for classifying legislative texts such as JRC-Acquis documents. Different approaches have been proposed for automatically assigning labels to JRC-Acquis documents [49, 135]. In most of these studies well-known classifiers have been combined with NLP techniques, such as part of speech tagging [141] and segmentation [47], to achieve a higher performance on JRC-Acquis dataset. Steinberger et al. [180] propose a framework called JEX for labeling documents with EuroVoc concepts. They first construct a profile for each EuroVoc concept and use the method proposed in [155] for learning a classifier.

2.3 Learning to Rank for Multi Label Text Classification

In this work, we use AdaRank [216] to learn the weights of the features. AdaRank learns $f(c, d)$, introduced in Section 2.2.2, from a collection of training examples. In our case, these are documents with their set of assigned labels. AdaRank optimizes the function $f(c, d)$ on the evaluation measure of Normalized Discounted Cumulative Gain (NDCG) over the complete ranked list.

Note that in this setup we are not only learning a model that *ranks* all classes given a document, but a function that *scores* classes given a document. In Section 2.4 we will re-estimate this scoring function based on co-occurrence patterns of the classes. Next, we describe the features used to construct $f(c, d)$.

2.3.1 Features for MLTC

We use different sources of information for extracting the features. The sources used for MLTC are: (1) labeled documents, (2) textual labels of classes, and (3) the relations between classes (thesaurus structure). We create different representations for documents and classes, and use them to extract features.

Representations of documents are based on both title and body text of documents. The first representation (*title representation*) is based on the titles of documents. We first remove stopwords from the titles and stem them. Then, we represent the titles as bags of stemmed unigrams. The second representation (*text representation*) is the bag of stemmed unigrams without stopwords based on all text of the document (including the title).

Similarly, we create four representations for each class c. The first two representations (*title and text representations*) are the union of the title representation and text representation of all documents labeled by c, respectively. The third representation (*label representation*) is the bag of stemmed unigrams (without stopwords) of the label c. In our dataset the mean and median number of tokens in the label representation of the classes are 2.12 and 2, respectively. The fourth representation (*ancestors label representation*) is the union of the label representations of all ancestors of the class c in the thesaurus hierarchy.

We now use the constructed representations and define different features. These representations lead to 8 possible combinations of a document and class representations (2 times 4). Moreover, for estimating the similarity of each combination, we employ three IR measures: (a) language modeling similarity based on KL-divergence using Dirichlet smoothing, (b) the same as (a) but using Jelinek-Mercer smoothing, and (c) Okapi-BM25. This leads to 24 features that are based on the textual similarity of documents and classes.

In addition to the features reflecting the textual similarity of documents and classes, we define a number of features reflecting the characteristics of classes independent of documents. First, the statistics of the classes in the training data is considered the prior knowledge for determining the likelihood of selecting a class as a label for documents. We define the number of times a class has been selected for annotating documents in the training data as its *popularity*. Second, the degree of ambiguity of a class implicitly affects its chance for being assigned to documents. We have used the relations between classes in the thesaurus hierarchy and modeled *ambiguity* with two different features: the number of parents of a class and the number of its children in the thesaurus graph. Another factor for determining the chance of a class of being selected as an annotation of a given document is its *generality*. We quantify the generality of a class as its depth in the thesaurus hierarchy (i.e., the length of its shortest path to the root).

Finally, for each document-class pair d and c, we construct a feature vector of size 28 (24 features based on the similarity of d and c, and 4 features based on the statistics of c). The value of each feature is normalized using Min-Max normalization and re-scaled to the [0, 1] interval.

2.4 Propagation Framework

In this section, we describe the score propagation framework for re-estimating the similarities of documents and classes based on the implicit relations between classes. This implicit information is the *co-occurrence* of classes in the labeled documents. Given a set of documents D and a set of classes C, let a similarity function $f(c, d)$ as in Section 2.2.2 be defined. We can represent this function as a $|C| \times |D|$ matrix S. We first normalize S by dividing each column S_d by the sum of its values, so that all columns add up to 1. Then, we will step-by-step re-estimate S by incorporating co-occurrence patterns of classes. For that, we create a conditional probability matrix P of size $|C| \times |C|$. For each class c, the row P_c, is defined as follows: for each class c'

$$P_c(c') = \begin{cases} P(c'|c), & \text{if there is a document labeled by both } c \text{ and } c' \text{ and } c \neq c' \\ 0, & \text{otherwise,} \end{cases}$$

where

$$P(c'|c) = \frac{|D_{c'} \cap D_c|}{|D_c|}, \tag{2.2}$$

and D_c is the set of documents labeled with class c.

Now let $S^0 = S$. We re-estimate the scores using P as follows, where t indicates the iteration:

$$S^t = \alpha S^{t-1} + (1 - \alpha) P S^{t-1}. \tag{2.3}$$

Here, α is the *neighborhood contribution* parameter controlling how much we smooth S with the co-occurrence matrix. After each iteration we normalize the values again by dividing each column by its sum.

This score propagation framework has two hyperparameters: α and the number of iterations t. In Section 2.6 we discuss their influence and determine their optimal values.

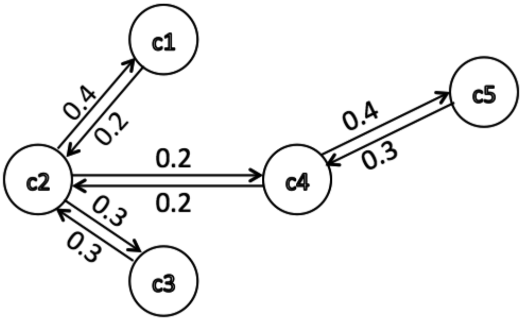


Figure 2.1: The graph of classes for matrix P introduced in Example 1.

Example 2.1: We give an example to illustrate how score propagation works. For simplicity, we assume that we want to re-estimate scores for only one document d. Assume that we have five classes ($c1, c2, ..., c5$) and the following P and S matrices:

$$P = \begin{bmatrix} 0 & 0.2 & 0 & 0 & 0 \\ 0.4 & 0 & 0.3 & 0.2 & 0 \\ 0 & 0.3 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0.4 \\ 0 & 0 & 0 & 0.3 & 0 \end{bmatrix}, S = \begin{bmatrix} 0.2 \\ 0.3 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix}$$

We can represent the matrix P as the directed graph shown in Figure 2.1, where nodes are classes and edges are weighted based on P. A conditional probability $P(c'|c) = w$ is represented as an edge from c to c' with weight w. In the first iteration of the algorithm, the score of $c2$ will be re-estimated based on its own score and the scores of $c1$, $c3$, and $c4$ as they are direct neighbors of $c2$ ($S[2] = \alpha S[2] + (1 - \alpha)(0.2S[1] + 0.3S[3] + 0.2S[4])$). The score of $c4$ will be re-estimated based on its own score and the scores of $c2$ and $c5$. In the second iteration, the score of $c2$, again will be affected by the re-estimated scores of its direct neighbors. Since the score of $c4$ is already affected by the score of $c5$, in the second iteration the score of $c2$ will be affected by the score of $c5$ as well. Therefore, in iteration t, the indirect neighbors that are reachable by t edges are used for re-estimating scores.

2.5 Experimental Setup

In this section, we describe our research questions, the data, the experiments, and the baselines with which we compare our proposed methods. We recall the two main research questions we want to address in this chapter.

RQ2.1 How effective is a LTR approach integrating a variety of sources of information as features for MLTC?

RQ2.2 Is it worthwhile to use the co-occurrence patterns of classes for MLTC?

We also perform a feature analysis. We first measure the relative importance of each feature and then see if we can create an almost optimal performing system using only a small set of features. We perform several analyses to understand how the score propagation works and what kind of documents benefit the most from it. Additionally, we examine the effectiveness of a dynamic thresholding method for selecting the number of classes for documents. Moreover, we study the effect of dataset size on the performance of different methods.

2.5.1 Dataset, pre-processing, and parameters

We use the English version of the JRC-Acquis dataset which contains documents of the European Union which are mostly on legal and political topics [179]. This dataset contains about 25,000 documents that have been manually labeled with EuroVoc concepts [63]. EuroVoc contains 6,796 hierarchically structured concepts, used to annotate political documents and news within the EU and in national governments. Since the structure of documents has changed over the years, we only use the documents of the last five years: from 2002 to 2006. We use the English version of JRC-Acquis, which contains 16,824 documents, each labeled with 5.4 concepts on average. The median and standard deviation of the number of labels per document are 5 and 1.83, respectively. Each document has a title, body text, and EuroVoc labels assigned to it. The mean and median length of titles of the documents are 19 and 18 words, respectively. The mean and median length of the texts are 2015 and 665 words, respectively.

To evaluate the proposed classification methods, we order the collection chronologically and train on the old documents and test on the newest documents. From an application perspective this is the most natural setup. The 70% oldest documents are used to construct the representations of classes (as documents in LTR) and the co-occurrence matrix of classes. The remaining 30% of the collection is used for training and testing. Naturally, the vocabulary and the set of classes used as labels differ in these two parts. To have a fair measure, we have removed the classes from the documents that do not occur in the first part. This leads to 1,639 different classes in our dataset. We do 5-fold cross validation on the newest part. To have a comparable evaluation, for 5-fold cross validation on our baselines, we add the first 70% part of the collection to the training data used in each fold and train their model. We have trained the ranking model using different LTR algorithms. Among them, AdaRank [216] has a slightly better performance and we report the results of this method.

We use a cut-off point of 5 and report Precision@5 as the main measure to evaluate different methods, since the median number of classes per document in our dataset is 5. Therefore, Precision@5 approximately could be considered as R-Precision as well. Moreover, we compute Recall@5, micro-averaged F-measure, and Mean Averaged Precision (MAP). In Section 2.6.3, we use an approach for dynamically selecting the number of classes for documents (instead of using a fixed cut-off point) and study its effect on the performance of different classifiers.

We compare our approach to two baselines: JEX and SVM. JEX is one of the state of the art systems developed for classifying JRC-Acquis documents [180]. Similar to feature selection approaches for text classification [165], this method first represents each class as a bag of keywords. The keywords are extracted using TF-IDF statistics. Similarly, documents are also represented as bags of keywords. Then, the similarity (cosine and BM25) of document and class representations are used to rank the classes with respect to the documents. For SVM, each document is represented by a feature vector using TF-IDF values. Each element of this vector corresponds to a word and its value is the TF-IDF weight of the word in the document normalized by the length of the document. Then, we train an SVM model to estimate the similarity of documents and classes and use these similarity scores to rank the classes with regard to the documents. In constructing the training set for SVM, we assume that a document belongs to all classes it is labeled with and add the document to the training material of those classes. The pre-processing done in this study is the same as in JEX: we employ the Porter stemmer and consider the 100 top frequent words in the collection as stopwords, which are removed.

Hyperparameter settings We use default parameter settings for JEX. These are optimal for the JRC-Acquis dataset. We use different parameters for the similarity functions used as features in LTR. Based on pilot experiments, when using titles of documents for calculating the similarities, we use these parameters: $\mu = 1,000$ for LM-Dirichlet, $\lambda = 0.2$ for LM-JM, and $b = 0.65$ and $k_1 = 1.2$ for Okapi BM25. When we use the text of documents for calculating the similarities we use these parameters: $\mu = 2,000$ for LM-Dirichlet, $\lambda = 0.6$ for LM-JM, and $b = 0.75$ and $k_1 = 1.2$ for Okapi BM25.

2.6 Experimental Results

In this section, we first answer our two research questions described in Section 2.5. Then, we study the impact of a dynamic thresholding method for selecting the number of classes for documents on the performance of different classifiers. Afterwards, we analyze the effect of training set size on the performance of the classifiers. Finally, we focus more on the score propagation method and study its impact on different types of documents.

2.6.1 Effectiveness of LTR

In this section, we evaluate how effective LTR is at integrating a variety of sources of information for MLTC and look at the importance of the different features. Table 2.1 shows the results of the LTR method compared to the baseline system and JEX in terms of Precision and Recall. BM25-TITLES ranks the classes based on the similarities of their title representation with the title representations of documents. This is the best performing single feature and is significantly better than JEX. Two observations can be made from Table 2.1. First, the LTR method significantly outperforms SVM, BM25-TITLES, and JEX, demonstrating that the additional sources of information employed in LTR are effective for the MLTC task. Second, the performance of SVM is the worst among all methods. This result shows that a standard classification approach is not effective for MLTC when there are very many classes. We do an additional experiment in which we use the scores estimated by JEX as an additional feature in the LTR approach. The performance of this approach is 0.5431 in terms of Precision and 0.5608 in terms of Recall. Adding JEX to the LTR approach improves the Precision

Table 2.1: Performance of SVM, JEX, best single feature, and LTR methods for MLTC. We report incremental improvement and significance over JEX($^\blacktriangle$ indicates t-test, one-tailed, p-value < 0.05)

Method	Precision (%Diff.)	Recall (%Diff.)	F1 (%Diff.)	MAP (%Diff.)
SVM	0.4146	0.4612	0.4366	0.4831
JEX	0.4353	0.4863	0.4505	0.5102
BM25-TITLES	0.4798 (10%)$^\blacktriangle$	0.5064 (4%)$^\blacktriangle$	0.4927 (9%)$^\blacktriangle$	0.5516 (8%)$^\blacktriangle$
LTR	0.5206 (20%)$^\blacktriangle$	0.5467 (12%)$^\blacktriangle$	0.5362 (19%)$^\blacktriangle$	0.6104 (20%)$^\blacktriangle$

of LTR by 4%. This result shows that JEX is also providing an additional informative feature.

We now look at the individual features used in the LTR model.

Importance of Different Information Sources for MLTC. We use the trained model of SVM-Rank [98] as well as the Precision of employing each individual feature for feature analysis. We assume the weight of each feature is a reflection of its importance. We use SVM-Rank for feature analysis because the weights it gives to the features are more smoothed. The performance of SVM-Rank is almost the same as the performance of AdaRank. Its performance in terms of Precision is 0.5013. AdaRank assigns very high weights to a few features and zero weights to the others. Here, we only want to analyze the importance of features and AdaRank's model does not reflect it very well.

Figure 2.2 illustrates the importance of a selected set of exploited features. We pick only one of the similarity methods (BM25) from each feature type since the other two get very similar scores.

Similarity of title representations of documents and classes is the best performing feature. The performance of this feature is significantly better than the performance of the feature defined using text representations of both classes and documents. Similarity of text representation of the given document and title representation of the classes is also an effective feature. Therefore, title representations can be considered as a succinct predictor of classes. JRC-Aquis is a set of political documents. Titles of political documents tend to be directly descriptive of the content, making the title the most informative part of the document. In addition to this, human annotators will pay considerable attention to the titles. Recall that the titles in this dataset are relatively long (the median is 18 words).

All query-independent features by themselves have very low Precision. Used together with the other features, generality and ambiguity get a very low weight in contrast to popularity. Investigating the hierarchy graph of the concepts, we see that there is little variation in generality: the average number of levels in the hierarchy is 3.85 and its standard deviation is 1.29. There is a considerable difference in the ambiguity: the average number of children is 4.94 (standard deviation is 4.96) and the average number of parents is 1.08 (standard deviation is 0.25). Ambiguity may have low importance because it is not discriminative on this data. The popularity feature is weighted high because of the skewness of the assigned class labels in JRC-Acquis [63].

The analysis both confirms the intuitions (e.g., the importance of labeled examples) but also highlights features of the document genre (e.g., the importance of titles), and human annotating behavior (e.g., the importance of popularity).

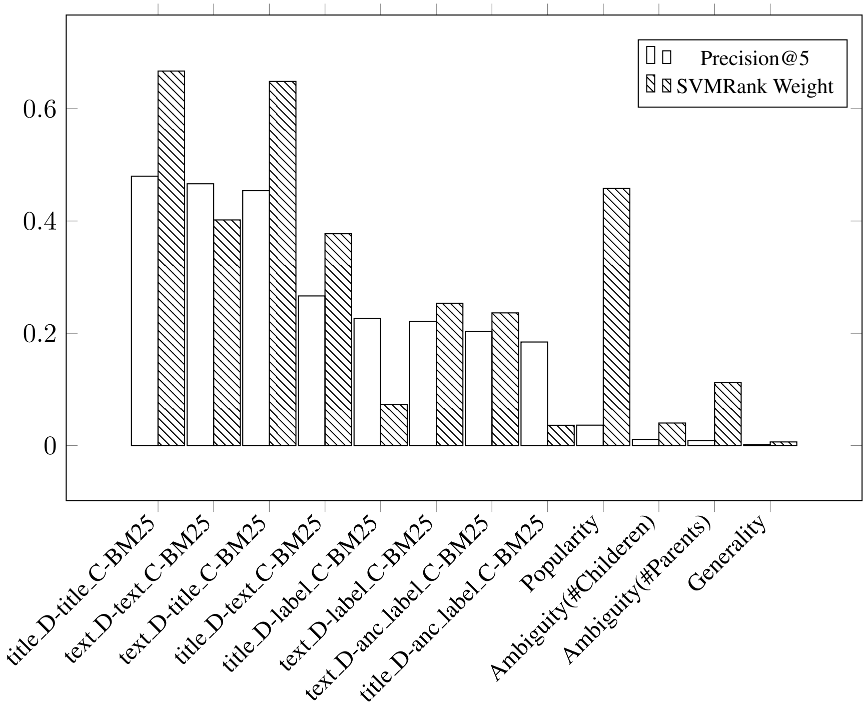


Figure 2.2: Feature importance: (1) Precision@5 of individual features, (2) weights in SVM-Rank model. $title\_D$ and $text\_D$ are title and text representations of document, respectively. $title\_C$, $text\_C$, $label\_C$, and anc_label_C are title, text, label, and ancestors' label representations of classes, respectively.

Lean and Mean Approach. The LTR approach uses a rather large set of features. Can we do almost as well with a small subset of the features?

We use our feature analysis to design a small system which uses a diverse combination of sources of information and employs the most efficient features from each source. Our lean-and-mean system is an LTR system trained on four selected features: the BM25 similarities of text representation of documents with text representation, title representation, and label representation of classes, and popularity of classes. Table 2.2 indicates the performance of this LTR-TTGP approach using only four features. The LTR-TTGP approach is significantly better than JEX and BM25-TITLES. Although the performance of LTR is significantly better than the LTR-TTGP method, the performance of LTR-TTGP is 96% of the large LTR system. Moreover, based on our analysis, the computation time of the selected features is less than 50% of the time needed for computing whole features. This makes the selective LTR approach a computationally attractive alternative to the full LTR approach.

2.6.2 Effectiveness of score propagation

In this section, we evaluate the use of co-occurrence patterns of classes in our classifiers. We will see that this method is indeed effective, although the gain diminishes for better

Table 2.2: Performance of LTR on all feature compared to four selected features (LTR-TTGP)

Method	Precision	Recall	F1	MAP
LTR	0.5206	0.5467	0.5362	0.6104
LTR-TTGP	0.5058	0.5301	0.5176	0.5947

Table 2.3: Performance of the score propagation approach. We report incremental improvement and significance of each score propagation approach over its non-propagated version ($^{\blacktriangle}$ indicates t-test, one-tailed, p-value < 0.05)

Method	Precision (%Diff.)	Recall (%Diff.)	F1	MAP
Propagated SVM	0.4758 (15%)$^{\blacktriangle}$	0.5023 (9%)$^{\blacktriangle}$	0.4880 (12%)$^{\blacktriangle}$	0.5356 (11%)$^{\blacktriangle}$
Propagated JEX	0.4912 (13%)$^{\blacktriangle}$	0.5246 (8%)$^{\blacktriangle}$	0.5073 (13%)$^{\blacktriangle}$	0.5483 (7%)$^{\blacktriangle}$
Propagated BM25-TITLES	0.5263 (10%)$^{\blacktriangle}$	0.5489 (8%)$^{\blacktriangle}$	0.5373 (9%)$^{\blacktriangle}$	0.5911 (7%)$^{\blacktriangle}$
Propagated LTR-TTGP	0.5334 (5%)	0.5593 (6%)	0.5460 (5%)	0.6203 (4%)
Propagated LTR	0.5470 (6%)$^{\blacktriangle}$	0.5719 (5%)$^{\blacktriangle}$	0.5591 (4%)$^{\blacktriangle}$	0.6320 (4%)$^{\blacktriangle}$

classifiers. Table 2.3 shows the results of propagating scores for the previously used approaches. In all cases we use the same hyperparameter setting: the neighborhood contribution parameter α is equal to 0.7 and the number of iterations of the propagation approach is 2. Re-estimating the scores using classes' co-occurrence patterns improves performance for all classifiers. Moreover, the results show that the propagation has the highest positive impact on SVM scores. SVM (without propagation) has the lowest performance among all methods. This indicates that when the classifier has a low quality, re-estimating the scores by propagation is more useful.

To gain additional insights into the score propagation approach, we analyze the effect of the parameters on the performance: α and the number of iterations. Figure 2.3 shows the Precision of different approaches for different values of α. The value of α has a great impact on the performance of all systems. The best performance is achieved for $0.7 \leqslant \alpha \leqslant 0.8$ which indicates that although the co-occurrence information is useful for annotating documents more accurately, the similarity of classes and documents is still much more important.

Figure 2.4 shows the Precision of different classifiers in different iterations of score propagation. All methods achieve their best performance with a few iterations. With only one iteration, the score of a class c is affected by the scores of its direct neighbors (a class c' is a neighbor of c if there is a document labeled by both c and c'). Therefore, a low number of iterations corresponds to using only the scores of co-occurring classes to re-estimate the score of c. With a high number of iterations, the score of c is affected by the scores of the classes that are indirect neighbors of c and might not be related to c. Therefore, the results are better with a few iterations where only close neighbors contribute to the score of a class and documents.

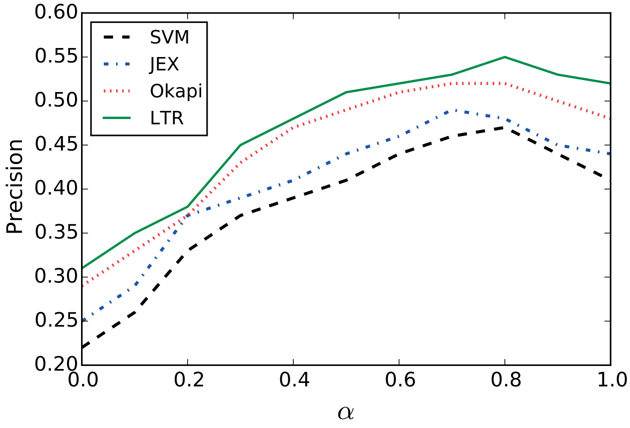


Figure 2.3: The effect of α parameter on Precision achieved by propagating the scores of different text classification approaches. The number of iterations is set to 2.

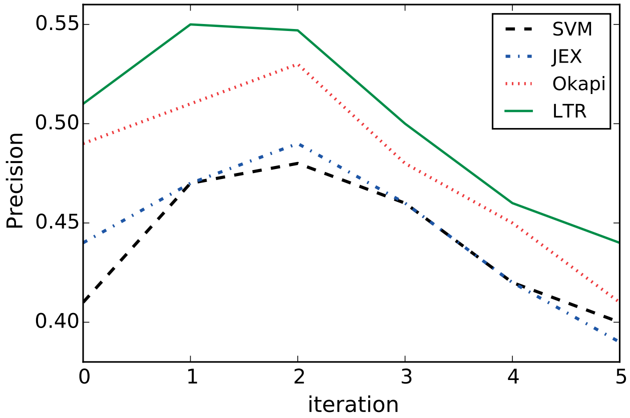


Figure 2.4: Precision achieved in different iterations of score propagation method for different text classification approaches. The value of α is set to 0.7.

2.6.3 The impact of using a dynamic threshold for choosing the number of classes

All the results reported so far are achieved by ranking classes for documents and choosing the top 5 classes as labels of documents. In other words, we use the median number of classes in the training set (which is 5) as the number of classes for every document. In this section, we relax this assumption and use a dynamic threshold for choosing the number of classes and study its impact of the performance in MLTC.

To choose the number of classes for documents, we use the approach proposed in [62, 219]. This method tries to learn a mapping from a ranked list of classes to its optimal number of classes using a training set. The training data for this approach is a set of ranked lists of classes (sorted by their scores regarding a document) and an optimal threshold for each ranked list which is achieved by minimizing a classification loss. Given this training set, the thresholding method tries to find a mapping from the space of ranked lists to the space of thresholds. After selecting a threshold for a document, classes with a score higher than the threshold are assigned to the document. As in [219], the classification loss is defined as the sum of false negatives and false positives. Therefore, the training set for learning the mapping is automatically constructed by using a ranker to rank classes regarding the documents and assigning a threshold to the ranked list that minimizes the classification loss. After creating this set, an optimal mapping is learned based on a linear-least-square-fit solution [219] using the following equation:

$$\min_{w^*,b^*} \sum_{i=1}^{m} ((w^* r(d_i) + b^*) - s(d_i))^2, \tag{2.4}$$

where m is the number of documents in the training set, $r(d_i)$ is a list of scores corresponding to the ranked list of classes with regards to a document d_i and $s(d_i)$ is the optimal threshold for $r(d_i)$ determined by minimizing the classification loss. The goal of the linear-least-square-fit is to determine the parameters of the linear mapping ($w*$ and $b*$) and use them to determine the thresholds for new instances. We use the same set of documents used for creating the classifier for learning the thresholding method. After having obtained the learned thresholding model, we use it to select the number of classes for documents in test set.

Table 2.4 shows the results of this experiment using different classifiers (note that we do not report MAP for this experiment as it is the same as the ones reported in Table 2.1). Dynamic thresholding improves performance on both Recall and Precision for all classifiers. The improvements are significant for Recall but not for Precision (see Table 2.4). Figure 2.5 shows the distribution of number of classes for documents in the dataset. The distribution is almost normal with a mean of 5. This explains why a static threshold of 5 has a reasonable performance. Based on the results, on the best performing method (Propagated LTR), for 89% of documents with more than 5 classes, the dynamic thresholding picks a number higher than 5. For documents with less than 5 classes, only in 69% of times a number less than 5 is picked. This gives us more insights on why Recall is improved more than Precision, as the dynamic thresholding in general tends to work better for documents with more classes.

Table 2.5 shows the root mean squared error (RMSE) and the mean absolute error (MAE) between the true number of classes and the estimated number of classes for different methods. Moreover, in this table, the accuracy of dynamic thresholding in choosing the correct number of classes for documents is also shown. The results indicate that in general dynamic thresholding performs better in terms of all metrics when the underlying classifier has a good performance. The main reason for this is that when the underlying classifier has a poor performance, the optimal threshold for the ranked list which is obtained by minimizing the classification loss is not reliable. The optimal performance for the thresholding method will be achieved with a perfect classifier which ranks all true classes on top of the ranked list, as in this case the optimal threshold will correspond to the actual number of classes. However, as the performance of the classifier degrades, the ranked lists used for creating the training set for thresholding get

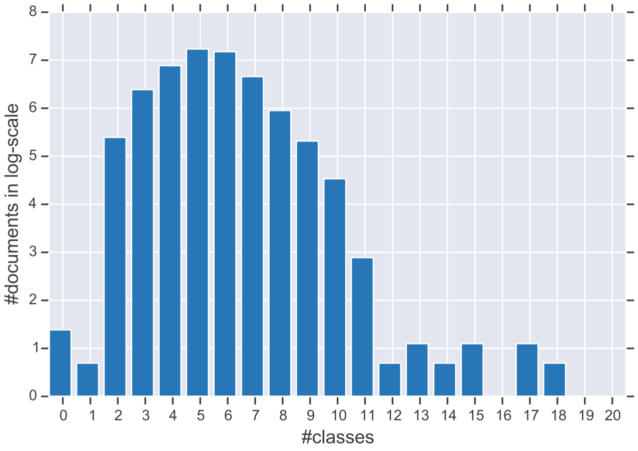


Figure 2.5: The distribution of number of classes in documents. X-axis corresponds to the number of classes assigned to documents in the ground-truth and Y-axis corresponds to the number of documents in log-scale.

Table 2.4: Performance of SVM, JEX, BM25-TITLES, and LTR, and Propagated LTR methods for MLTC using a dynamic threshold for selecting the number of classes. The significance tests are done on the improvements of each method using a dynamic threshold over its corresponding method which uses a static threshold, e.g., setting number of classes to 5.

Method	Precision (%Diff.)	Recall (%Diff.)	F1 (%Diff.)
SVM	0.4201 (1%)	0.4816 (4%)$^\blacktriangle$	0.4487 (3%)$^\blacktriangle$
JEX	0.4424 (2%)	0.5012 (3%)$^\blacktriangle$	0.4699 (4%)$^\blacktriangle$
BM25-TITLES	0.4874 (2%)	0.5194 (3%)$^\blacktriangle$	0.5028 (3%)$^\blacktriangle$
LTR	0.5248 (1%)	0.5687 (4%)$^\blacktriangle$	0.5459 (2%)$^\blacktriangle$
Propagated LTR	0.5510 (1%)	0.5952 (4%)$^\blacktriangle$	0.5722 (2%)$^\blacktriangle$

more unreliable.

Overall, based on the results in this section, we conclude that dynamic thresholding is an effective approach for selecting the number of classes and it can have a big impact of the performance of the classifier in MLTC task.

2.6.4 The impact of dataset size on the performance of different classifiers

In this section, we study the impact of size of training samples per class on the performance of different classifiers. To do this, we select the classes that have at least 20 training samples. Afterwards, we balance the dataset by taking an equal number of instances per class. Then, we vary the number of samples per class and report the performance of using different number of instances per class. Figure 2.6 shows the results of this experiment. This subset of the dataset contains 4381 classes. Again, we use the 70% oldest documents for creating the representation of classes and the

Table 2.5: The root mean squared error (RMSE) and mean absolute error (MAE) between the assigned number of classes and the actual number of classes for documents and the accuracy of the thresholding method in choosing the correct number of classes for documents for SVM, JEX, BM25-TITLES, and LTR, and Propagated LTR methods for MLTC. The accuracy is calculated by dividing the number of documents for which the thresholding method picks a correct number of classes by total number of documents. We also report RMSE, MAE, and accuracy for the fixed threshold method (choosing top 5 classes)

Method	RMSE	MAE	Accuracy
Fixed threshold	1.87	1.39	0.23
SVM	1.62	1.21	0.26
JEX	1.51	1.07	0.28
BM25-TITLES	1.47	0.99	0.30
LTR	1.33	0.83	0.34
Propagated LTR	1.26	0.75	0.36

remaining 30% for training and testing the models.

The performance of the LTR method is better than the performance of other classifiers for different number of samples per class. This result, again, indicates that the LTR method is the most effective method for MLTC. As the results show, the performance of all methods is improved by increasing the number of samples per class. However, the rate of the improvement is different for different models. The rate is quite high for SVM and this method can benefit more from more samples. This is expected as adding more samples helps SVM learn a better decision boundary for classes and generalize better. This effect is similar for JEX and this method also benefits a lot from more samples. The LTR method achieves a good performance with 100 examples per class and the performance does not change much by increasing the number of instances. The impact of increasing the number of samples on the performance of BM25-TITLES is similar to this effect on LTR. Both LTR and BM25-TITLES are ranking-based classifiers. This results indicate that, first, ranking-based methods are more stable with regard to the number of samples than classification approaches. These models create a profile for each class and even with a few samples the created profiles are good enough for computing the similarity of classes and documents. Second, LTR that tries to combine different sources of information is the most stable method. This shows the impact of using all available information on the performance in MLTC. An intelligent combination model can achieve a reasonable performance even when there are only a few samples per class.

2.6.5 What kind of documents benefit the most from score propagation?

In this section, we analyze the effect of score propagation on the performance achieved for different types of documents. To do this, we first bin the documents based on the number of classes assigned to them in the ground-truth data. Then, we use the LTR method to assign classes to the documents and measure the Precision for each bin. Our running hypothesis is that the score propagation method should have a better

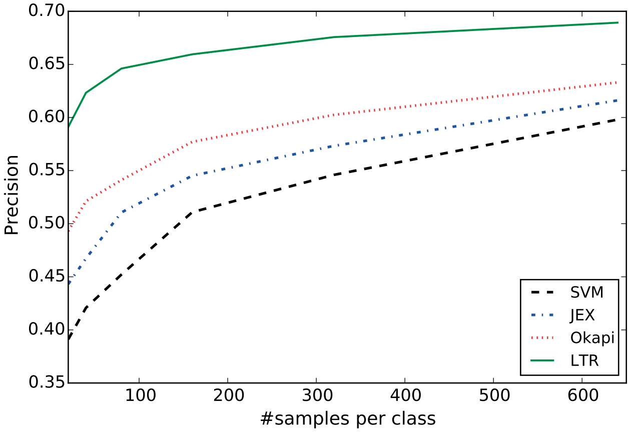


Figure 2.6: Precision achieved using different sample sizes for classes. X-axis corresponds to the number of samples used per class for training the classifiers.

performance on documents with more classes, as in this case, there is more information from neighboring classes available for assigning classes.

Figure 2.7 shows the results of this experiment. The results show that with a small number of iterations, the performance for documents with a small number of classes is higher. For example, for documents with only three or four classes, the best performance is achieved when we do not use the score propagation method at all (Iteration=0). In this case, increasing the propagation effect (number of iterations) results in a lower Precision. On the other hand, for documents with a high number of classes (documents with more than six classes), by increasing the number of iterations, the Precision keeps getting improved.

These results indicate that when there is more information provided by neighboring classes, the score propagation method can exploit it to refine the assignment of classes to the documents more accurately. Our score propagation method has an exploratory behavior. With a small number of iterations the exploration effect is low. However, when we increase the number of iterations it tries to explore more and use the information provided by indirect neighbors. The results presented in this section indicate that this exploration has a positive impact on more ambiguous documents (documents with a high number of classes) and getting more information from even indirect neighbors can help disambiguate these kind of documents. For less ambiguous documents, the score propagation method is not helpful indicating that there is no need for exploration for these kind of documents. Therefore, we conclude that it is best to use the score propagation method when there is a need for exploration and documents have a high number of classes.

2.7 Conclusion

Simple classification approaches (such as binary classification methods) for Multi-Label Text Classification (MLTC) fail when there is a large number of classes. In this chapter,

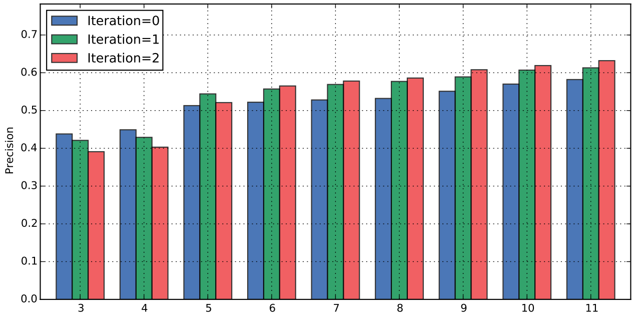


Figure 2.7: Precision achieved for different bins of documents based on their actual number of classes in the ground-truth data. X-axis corresponds to the number of classes per document.

we considered MLTC as a ranking problem and proposed Learning to Rank (LTR) as a solution. Our approach is based on combining different sources of information for MLTC. We found that LTR can effectively exploit several sources of evidence, leading to significant improvements over the state of the art. The LTR approach is rather brute force, but is able to infer many complexities of human class assignment based on the observed data. It can also be viewed as a (soft) upperbound on performance, also taking into account the inter-indexer agreement, e.g., [88]. Our findings mostly confirmed the intuitions with the labeled examples as the key source of evidence. The title evidence was remarkably important, due to its descriptive nature and high precision. Interestingly, popularity is a feature without any power in isolation, but very effective in combination in order to capture important aspects of human labeling behavior.

We proposed to use the co-occurrence patterns of classes in the labeled documents to improve the accuracy of the MLTC classifier. This is more effective when the underlying classifier has a low accuracy, indicating that co-occurrence patterns of classes are important signals for classifying documents in the MLTC task.

The findings of this research have several theoretical implications. First, the fact that the ranking based MLTC classifiers perform better than traditional classification approaches implies that more effective MLTC systems can be designed by defining and optimizing for a ranking loss rather than a classification loss. Second, rather than just using a set of training samples, utilizing various sources of information leads to better performances in MLTC. LTR is an effective approach for unifying and utilizing different information sources for MLTC. An interesting future research direction can be designing MLTC systems using various sources of information and by adapting a ranking loss into the MLTC framework. The findings of this research also have practical implications on designing MLTC systems. The identification of a subset of effective features from all sources of information opens up the possibility to design efficient MLTC systems. Also, our analysis on the importance of different features can help human annotators to concentrate their focus on the important parts of documents while assigning labels to them. Moreover, our findings imply that it is important to use other sources of information such as co-occurrence patterns of classes for hard classification problems. The designed classifier can be adapted in various applications such as exploratory search, automatic indexing of textual documents, text summarization, and mapping text collection to ontologies.

In this chapter, we studied how metadata and structure can be helpful in classifying topically similar documents together. In the next chapter, we take a different angle for classifying documents and study how we can measure topical diversity of documents and use it for classifying them.

3 Hierarchical Topic Model Re-estimation for Measuring Topical Diversity

In this chapter, we propose a method for measuring topical diversity of documents and use it to classify documents. We analyze different aspects of the proposed diversity measure and answer **RQ2**.

3.1 Introduction

Quantitative notions of measuring topical diversity of text documents are useful in a number of applications, such as assessing the interdisciplinariness of a research proposal [15] and helping to determine the interestingness of a document [7, 61].

Well over three decades ago, an influential formalization of diversity was introduced in biology [162]. It decomposes diversity in terms of three central concepts: *elements* that belong to *categories* within a *population* [177]. Given a set T of categories that partitions a population d, the diversity of d is then defined as

$$div(d) = \sum_{i \in T} \sum_{j \in T} p_i^d p_j^d \delta(i, j), \tag{3.1}$$

where p_i^d denotes the proportion of category i in d and $\delta(i, j)$ is the distance between categories i and j, which can be calculated in a chosen manner. This notion of population diversity can be interpreted as the expected distance between two randomly selected (with replacement) elements of the population.

Bache et al. [15] have adapted the biological notion of population diversity to quantify the topical diversity of a text document. For measuring the topical diversity of a text document, words are considered elements, topics are categories, and a document is a population. When using topic modeling for measuring topical diversity of a text document d, Bache et al. [15] estimate elements based on the probability of a word given the document ($P(w \,|\, d)$), categories based on the probability of a word given a topic ($P(w \,|\, t)$), and populations based on the probability of a topic given the document ($P(t \,|\, d)$).

This chapter was published as [10].

In probabilistic topic modeling, at estimation time, these distributions are usually assumed to be sparse. First, the main content of documents is assumed to be generated by a small subset of words from the vocabulary (i.e., $P(w\,|\,d)$ is sparse). Second, each topic is assumed to contain only some topic-specific related words (i.e., $P(w\,|\,t)$ is sparse). Finally, each document is assumed to deal with a few topics only (i.e., $(P(t\,|\,d)$ is sparse). When approximated using currently available methods, however, $P(w\,|\,t)$ and $P(t\,|\,d)$ often turn out to be dense rather than sparse [129, 176, 195]. Dense distributions cause two important problems for the quality of topic models: *generality* and *impurity*. General topics mostly contain general words. They are typically assigned to most of the documents in a corpus. In other words, the $P(t\,|\,d)$ distributions are not document-specific. Impure topics contain words that are not related to the topic. These impure words are mostly general words. Generality and impurity of topics are problematic when estimating topical diversity of text documents since they both result in low quality $P(t\,|\,d)$ distributions. Recall that these are core to the topical diversity score based on the biological notion of diversity (Equation 3.1).

To improve the measurement of topical diversity of text documents we propose a hierarchical way of making the three distributions $P(w\,|\,d)$, $P(w\,|\,t)$ and $P(t\,|\,d)$ more sparse. To this end we re-estimate the parameters of these distributions so that general, collection-wide items are removed and only salient items are kept. For the re-estimation, we use the concept of parsimony [80] to extract only essential parameters of each distribution.

We evaluate the performance of the proposed hierarchical re-estimation method for measuring topical diversity of text documents and compare our approach against the state-of-the-art [176]. In doing this, we answer **RQ2**:

> *How effective is our hierarchical re-estimation approach in measuring topical diversity of documents? Are the topic models that have been improved in this way also successfully applicable for other tasks such as documents clustering and classification?*

Our main contributions are: (1) We propose a hierarchical re-estimation process for topic models to address the two main problems in estimating the topical diversity of text documents, using a biologically inspired definition of diversity. (2) We study each level of re-estimation individually in terms of efficacy in solving the general topics problem, the impure topics problem, and improving the accuracy of estimating the topical diversity of documents. (3) We study the impact of re-estimation parameters on the statistics of documents and its relation to the quality of trained topic models and recommend effective settings of these parameters.

As an additional contribution, we also make the source code of our topic model re-estimation method available to the research community to further advance research in this area.[1]

[1] The source codes are available here: `https://github.com/HoseinAzarbonyad/HiTR`

3.2 Related Work

Our hierarchical topic model re-estimation touches on research in multiple areas. We review work in four directions: improving the quality of topic models, measuring text diversity, evaluating topic models, and parsimonization.

3.2.1 Improving the quality of topic models

Topic models are effective for modeling text documents and expressing the contents of text documents in a low-dimensional space [24]. Although topic models like Latent Dirichlet Allocation (LDA) are powerful tools for modeling data in an unsupervised fashion, they suffer from different issues, especially when dealing with noisy data [30]. As mentioned already, the two most important issues with topic models are the *generality problem* and the *impurity problem* [30, 129, 176, 195]. These problems with topic models have a negative influence on the performance of tasks in which topic models are applied besides document diversity, namely document clustering, document classification, document summarization, information retrieval, sentiment analysis (see [30] for an overview).

Wallach et al. [195] propose asymmetric Dirichlet priors to construct a general topic and assign general terms to this general topic in the learning process. Similar ideas to improve the quality of topic models have been employed by others [199, 209]. Similar to [195, 199, 209], one of our goals is to address the generality problem. The main difference, however, is that they do not aim to address the two issues mentioned with topic models directly and the topic representations and topic word distributions that they arrive at are neither parsimonious nor sparse. That is, in their approach, each topic could still have a non-zero assignment probability to each document. We hypothesize that parsimony is essential in topic modeling, since it is expected that each document only focuses on a few topics [176] and in contrast to the work cited above our goal is to achieve this parsimony.

Soleimani and Miller [176] propose parsimonious topic models (PTM) to address the generality and impurity problems. A shared topic is created and general words are assigned to this topic. PTM achieves state-of-the-art results compared to existing topic models. We also address the generality and impurity problems with topic models. The background language model in our model and the shared topic in PTM have similar functionalities. They are both used to handle and remove generality from topic-word distributions. However, in PTM, the shared topic is more complicated as for each word there are a few more parameters to be estimated: (1) whether a word is topic-specific for each topic and (2) probability of being topic-specific under each topic for each word. In our approach, we model all this using a background language model with far fewer parameters. Moreover, we model and remove the generality in at three different levels: document-word distribution, topic-word distribution, and document-topic distribution. PTM handles the generality in topic-word and document-topic distributions and does not handle the generality in document-word distributions explicitly.

3.2.2 Evaluating topic models

Topic models are usually evaluated either intrinsically, for example, in terms of their generalization capabilities, or extrinsically in terms of their contribution to external tasks [196]. We focus on extrinsic evaluations of the effectiveness of our re-estimation approach. Our main evaluation concerns its effectiveness in measuring the topical diversity of text documents. In addition, in Section 3.7, we analyze the effectiveness of our re-estimation approach in removing impurity from documents in terms of purity in document clustering and document classification tasks.

Specifically, in the document classification task, topics are used as features of documents with values $P(t\,|\,d)$. These features are used for training a classifier [112, 146, 176]. In the document clustering task, each topic is considered a cluster and each document is assigned to its most probable topic [146, 213, 218]. For the analyses in Section 3.7, following common practice (e.g., [119, 133, 146]), we use Purity and Normalized Mutual Information in the clustering task, and Accuracy as our prime evaluation metric in the classification task. Furthermore, the quality of topic models can be measured by the quality of the term distributions per topic, in terms of topic coherence [119, 146], and by having their interpretability judged by humans [35, 144].

3.2.3 Text diversity and interestingness

Prior to Bache et al. [15], measuring topical diversity of documents had not been studied comprehensively from a text mining perspective. Bache et al. [15] use Rao's diversity score (Equation 3.1) [162] to quantify diversity of text documents by means of LDA topic models [24]. In their framework, the diversity of a document is proportional to the number of dissimilar topics it covers. Similar to [15], Derzinski and Rohanimanesh [61] define the diversity of documents by means of topic models, but instead of Rao's measure they use an information theoretic diversity measure based on the Kullback Leibler divergence. Azarbonyad et al. [7] also use Rao's diversity measure to quantify the diversity of political documents and analyze the correlation of topical diversity and interestingness over political documents. Their main finding, however, is different from Derzinski and Rohanimanesh [61]'s conclusion, as they conclude that although in general topical diversity and interestingness of political documents are somehow correlated, a text's topical diversity does not necessarily reflect its interestingness.

3.2.4 Model parsimonization

Parsimonization refers to the process of extracting essential elements of a distribution and removing superfluous, general information. Parsimonization can be considered an unsupervised feature selection approach. The idea is to extract features containing information about samples and remove features that are not informative for explaining the samples [42, 120]. Because our hierarchical re-estimation process builds on parsimonious language models (PLMs) [80], we briefly review them.

PLMs were introduced in an information retrieval setting, in which language models are used to model documents as distributions over words. The goal of parsimonization in this context is to extract words that reflect the content of documents and remove collection-specific general words. To extract salient document-specific words for each document, some studies define a layered language model of documents where the language model of a document is composed of a general background model and a document-specific language model [223, 228]. The Expectation-Maximization (EM) algorithm is employed to estimate the parameters of such models. Using this idea, Hiemstra et al. [80] propose a method for parsimonizing document language models with the aim of removing general words by pushing the probabilities of the words that are well explained by the background model toward zero. We employ this approach for re-estimating and refining topic models.

Here we briefly recall the formal principles underlying PLMs. The main assumption is that the language model of a document is a mixture of its own specific language model and the language model of the collection:

$$P(w \,|\, d) = \lambda P(w \,|\, \tilde{\theta}_d) + (1 - \lambda) P(w \,|\, \theta_C), \tag{3.2}$$

where w is a term, d a document, $\tilde{\theta}_d$ the document specific language model of d, θ_C the language model of the collection C, and λ is a mixing parameter ($0 \leqslant \lambda \leqslant 1$). The main goal is to estimate $P(w \,|\, \tilde{\theta}_d)$ for each document. Language model parsimonization is an iterative EM algorithm in which the initial parameters of the language model are the parameters of the standard language model, estimated using maximum likelihood:

Initialization:

$$P(w \,|\, \tilde{\theta}_d) = \frac{tf_{w,d}}{\sum_{w' \in d} tf_{w',d}},$$

where $tf_{w,d}$ is the frequency of w in d. The following steps are done in each iteration of the algorithm:

E-step:

$$e_w = tf_{w,d} \cdot \frac{\lambda P(w \,|\, \tilde{\theta}_d)}{\lambda P(w \,|\, \tilde{\theta}_d) + (1 - \lambda) P(w \,|\, \theta_C))}, \tag{3.3}$$

M-step:

$$P(w \,|\, \tilde{\theta}_d) = \frac{e_w}{\sum_{w' \in d} e_{w'}}, \tag{3.4}$$

where $\tilde{\theta}_d$ is the parsimonized language model of document d, which is initialized by the language model of d, C is the background collection, $P(w \,|\, \theta_C)$ is estimated using maximum likelihood estimation, and λ is a parameter that controls the level of parsimonization. A low value of λ will result in a more parsimonized model while $\lambda = 1$ yields a model without any parsimonization. The E-step gives high probability values to terms that occur relatively more frequently in the document than in the background collection, while terms that occur relatively more frequently in the background collection get low probability values. In the M-step the parameters are normalized to form a probability distribution again. After this step, terms that receive a probability lower than a predefined *threshold* are removed from the model. The EM process will stop after a fixed number of iterations or when the models $\tilde{\theta}_d$ do not change significantly anymore.

PLM is a two-topic mixture model (the graphical model is shown in Figure 3.1, as can be seen θ_C is considered as an external observation and the goal is to estimate $\tilde{\theta}_d$

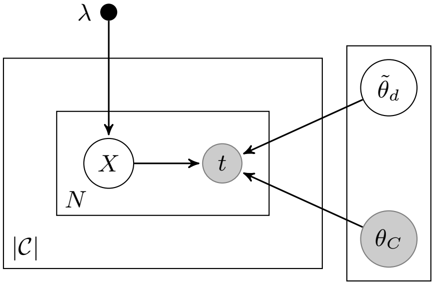


Figure 3.1: Plate diagram of PLM. X corresponds to e_w in Equation 3.3.

given θ_C and λ). In that sense, PLM is similar to an LDA model with two kinds of topics (general and specific topics). However, its mechanism is different than LDA. In LDA, all topics are shared among documents and only the proportions of topics (document-topic distributions) are different for different documents. In PLM, there is a general topic which is shared among all documents, but there is a specific topic for each document which is not shared with other documents. Moreover, in PLM, the λ controls the proportion of general and specific topics in documents and it is fixed.

3.3 Hierarchical Topic Model Re-estimation

In this section, we describe HiTR (*hi*erarchical *to*pic model *r*e-estimation). HiTR can be applied on top of any topic modeling approach that has two main components, $P(w\,|\,t)$ and $P(t\,|\,d)$ distributions.

3.3.1 Overview

The input of HiTR is a corpus of text documents. The output is a probability distribution over topics for each document in the corpus.

As explained in Section 3.1, the quality of topic models such as LDA is highly dependent on the quality of the $P(w\,|\,d)$, $P(w\,|\,t)$, and $P(t\,|\,d)$ distributions. However, generality and impurity of these distributions cause the poor quality of topic models. To solve these issues, we propose to apply re-estimation at three levels:

document re-estimation (DR) re-estimates the language model per document $P(w\,|\,d)$,

topic re-estimation (TR) re-estimates the language model per topic $P(w\,|\,t)$, and

topic assignment re-estimation (TAR) re-estimates the distribution over topics per document $P(t\,|\,d)$.

Based on applying or not applying re-estimation at different levels, there are 8 possible re-estimation approaches. Figure 3.2 gives a graphical overview of the different levels of re-estimation and how they are combined. HiTR refers to the model that uses all three re-estimation techniques, i.e., DR+TR+TAR that can be applied to any topic model TM.

To summarize, HiTR works as follows: we first do the DR step, then train a topic model (TM step) on top of the re-estimated documents. Afterwards, we apply the TR step on the trained topic model and use the re-estimated topic model (the topic model achieved after the TR step) to assign topics to documents. Finally, we apply the TAR

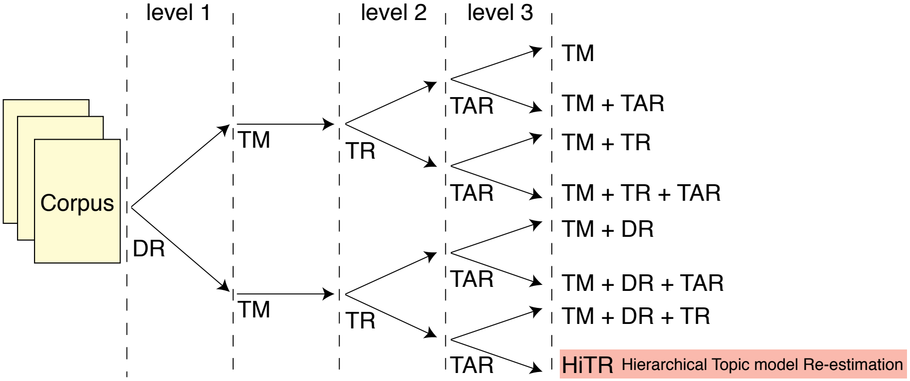


Figure 3.2: Different topic re-estimation approaches. TM is a topic modeling approach like, e.g., LDA. DR is document re-estimation, TR is topic re-estimation, and TAR is topic assignment re-estimation.

step to topics assigned to the documents using the re-estimated topic model. We follow this order of re-estimation for two reasons: first, for the topical diversity task we only use the document-topic distributions. And second, this order provides the maximum amount of re-estimation in the final document-topic distribution because at each step of re-estimation impurity and generality is removed from document-word and topic-word distributions and finally the remaining impurity and generality is removed using TAR. Next, we describe each of the re-estimation steps in more detail.

3.3.2 Document re-estimation

The first level of re-estimation is *document re-estimation* (DR), which re-estimates $P(w\,|\,d)$. The main intuition behind this level of re-estimation is to remove unnecessary information from documents before training topic models. This is comparable to pre-processing steps such as removing stopwords and high and low frequency words, that are typically carried out prior to applying topic models [23, 24, 119, 133, 146]. Proper pre-processing of documents, however, takes lots of effort and involves tuning several parameters, such as the number of high-frequent words to remove, if stopwords should be removed or not, whether rare words should be removed or not, whether IDF values should be considered in removing general/rare words. When dealing with a large document collection, finding optimum values for all of these parameters is non-trivial, while blindly removing words from documents without considering the distribution of them over documents could lead to missing important words and losing important information.

To solve this issue and pre-process documents automatically, we propose *document re-estimation*. After document re-estimation, we can train any standard topic model on the re-estimated documents. If general words are absent from (re-estimated) documents, we expect that the trained topic models will not contain general topics. Moreover, document re-estimation removes impure elements (general words) from documents, which will lead to more pure topics. Hence, document re-estimation is expected to address both the general topic and the impure topic problem.

Document re-estimation uses the parsimonization method described in Section 3.2.4. The parsimonized model $P(w \,|\, \tilde{\theta}_d)$ in Equation 3.4 is used as the language model of document d, and after removing unnecessary words from d, the frequencies of the remaining words (words with $P(w \,|\, \tilde{\theta}_d) > 0$) are re-calculated for d using the following equation:

$$tf_{w,d} = \left\lfloor P(w \,|\, \tilde{\theta}_d) \cdot |d| \right\rfloor,$$

where $|d|$ is the document length in words. Topic modeling is then applied on the recalculated document-word frequency matrix.

3.3.3 Topic re-estimation

The second level of re-estimation is *topic re-estimation* (TR), which re-estimates $P(w \,|\, t)$ by removing general words from it. The re-estimated distributions from this step are used to assign topics to documents.

The goal of this step is to increase the purity of topics by removing general words that have not yet been removed by document re-estimation. It is known from the literature [30, 129, 176, 195] that some topics extracted by means of topic models are impure and contain general words.

The two main advantages of applying TR are that (1) it results in more pure topics which are more interpretable by human, and (2) after getting pure, topics are less likely to be assigned to documents erroneously.

A topic is modeled as a distribution over words, which is itself a language model. Our main assumption is that each topic's language model is a mixture of its topic-specific language model and the language model of the background collection. The goal of TR is to extract a topic-specific language model for each topic and remove the part which can be explained by the background model. Given a set of topics T, background language model θ_T, and for each $t \in T$, a topic-specific language model $\tilde{\theta}_t$, we initialize $P(w \,|\, \tilde{\theta}_t)$ and $P(w \,|\, \theta_T)$ as follows:

$$\begin{aligned} P(w \,|\, \tilde{\theta}_t) &= P(w \,|\, \theta_t^{\mathcal{TM}}) \\ P(w \,|\, \theta_T) &= \frac{\sum_{t \in T} P(w \,|\, \theta_t^{\mathcal{TM}})}{\sum_{w' \in V_T} \sum_{t' \in T} P(w' \,|\, \theta_{t'}^{\mathcal{TM}})}, \end{aligned}$$

where $P(w \,|\, \theta_t^{\mathcal{TM}})$ is the probability of w belonging to topic t estimated by a topic model $\mathcal{TM}$, and V_T is the set of all words occurring in all topics. Having these estimations, the steps of TR are similar to the steps of PLM, except that in the E-step we estimate $tf_{w,t}$ (the frequency of w in t) using $P(w \,|\, \theta_t^{\mathcal{TM}})$.

3.3.4 Topic assignment re-estimation

The third and final level of re-estimation is *topic assignment re-estimation (TAR)* which re-estimates $P(t \,|\, d)$.

In topic modeling, most topics are usually assigned with a non-zero probability to most of documents. When documents are typically focused on just a few topics, this is an incorrect assignment, as topics should only be assigned to documents that deal with them. General topics assigned to a majority of documents are uninformative. The goal of TAR is to address the general topics problem and achieve more document specific topic assignments.

To re-estimate topic assignments, a topic model is first trained on the document collection. This model is used to assign topics to documents based on the proportion of words in common between them. We then model the distribution over topics per document as a mixture of its document-specific topic distribution and the topic distribution of the entire collection. The goal of TAR is to extract the document-specific topic distribution for each document and remove general collection-wide topics from them.

We initialize the document-specific topic distribution $P(t\,|\,\tilde{\theta}_d)$ and the distribution of topics in the entire collection C, $P(t\,|\,\theta_C)$ as follows:

$$\begin{aligned} P(t\,|\,\tilde{\theta}_d) &= P(t\,|\,\theta_d^{\mathcal{TM}}) \\ P(t\,|\,\theta_C) &= \frac{\sum_{d \in C} P(t\,|\,\theta_d^{\mathcal{TM}})}{\sum_{t' \in T} \sum_{d' \in C} P(t'\,|\,\theta_{d'}^{\mathcal{TM}})}. \end{aligned}$$

Here $P(t\,|\,\theta_d^{\mathcal{TM}})$ is the probability of assigning topic t to document d estimated by the topic model $\mathcal{TM}$. The remaining steps of TAR follow the ones of PLM. The only difference is that in the E-step, we estimate $tf_{t,d}$ using $P(t\,|\,\theta_d^{\mathcal{TM}})$.

3.4 Evaluating HiTR

To evaluate the performance of our approach to topical diversification, we follow the evaluation setup introduced in [15]. Our aim is to answer our second research question:

RQ2 How effective is our hierarchical re-estimation approach in measuring topical diversity of documents? Are the topic models that have been improved in this way also successfully applicable for other tasks such as documents clustering and classification?

To address **RQ2** we run our models on a binary classification task. We generate a synthetic dataset of documents with high and low topical diversity (the process is detailed in Section 3.5.2), and the task for every model is to predict whether a document belongs to the high or low diversity class. We employ HiTR to re-estimate topic models and use the re-estimated models for measuring topical diversity of documents. We compare our method to LDA (as also used in [15] for the same purpose) and to the state-of-the-art parsimonious topic models PTM [176]. The results of our experiments regarding **RQ2** are discussed in Section 3.6.1. Moreover, we evaluate the performance of HiTR in document clustering and classification tasks and analyze its effectiveness in these tasks. The results of these experiments are described in Section 3.7.

Additionally, to gain deeper insights into how HiTR performs, we conduct a separate analysis of each level of re-estimation, DR, TR and TAR and answer the following research questions:

RQ2.1 What is the effect of DR on the quality of topic models? Can DR replace manual pre-processing?

RQ2.2 Does TR increase the purity of topics? And if so, how does using the more pure topics influence the performance in topical diversity task?

RQ2.3 How does TAR affect the sparsity of document-topic assignments? And what is the effect of the achieved parsimonized document-topic assignments on the topical diversity task?

RQ2.1 concerns the effectivenes of DR in removing general words from documents and its effect on the quality of topic models. To answer **RQ2.1**, we train LDA models with and without manual pre-processing and with and without DR. We compare the quality of models achieved using different combinations. This will show how effective DR is in pre-processing documents automatically. Moreover, we measure corpus statistics such as vocabulary size, average type-token ratio, average document length after running DR with different parameters. We train LDA models on the corpora achieved with different parameters and measure the quality of trained models. Then, we analyze the correlation of corpus statistics achieved from DR with different parameters and the quality of models trained on them. In Section 3.6.2, the results regarding **RQ2.1** are described.

To answer **RQ2.2**, we first evaluate the performance of TR on the topical diversity task and compare its performance to DR and TAR. We focus on its effectiveness in removing impure words from topics and perform a qualitative analysis on topic models before and after running TR. The results of experiments regarding **RQ2.2** are discussed in Section 3.6.2.

To answer **RQ2.3**, we first evaluate TAR together with LDA in a topical diversity task and analyze its effect on the performance of LDA to study how successful TAR is in removing general topics from documents. The results of this experiment are presented in Section 3.6.2.

3.5 Topical Diversity with HiTR

In this section, we discuss the experimental setup for the topical diversity test.

3.5.1 Topical Diversity Measure

After re-estimating word distributions in documents, topics, and document-topic distributions using HiTR, we use the final distributions over topics per document for measuring topical diversity. Diversity of texts is computed using Rao's coefficient (Equation 3.1). For each topic x, observed in corpus C, we construct a vector V_x of length $|C|$ (the number of documents in the corpus). Each entry of this vector corresponds to a document d_y and its value is assigned as: $V_x[y] = p_x^y$. We use the normalized angular distance for measuring the distance between topics, since it is a proper distance function [7]:

$$\delta(i,j) = \frac{ArcCos(CosineSim(V_i, V_j))}{\pi},$$

where $CosineSim(\cdot,\cdot)$ is the cosine similarity between two vectors, and $ArcCos(\cdot)$ is the arc cosine. We use the distributions over topics per document for calculating the distance between topics. There are two possible approaches for measuring the topic distance: based on document-topic distributions or topic-word distributions. From a diversity perspective, document-topic distributions are more suitable for this task. For example, consider two topics that co-occur frequently in documents but have different topic-word distributions. In principle, if a document contains these topics, it should not be diverse, but since the topic-word similarity of these two topics is low the document will have a high diversity.

3.5.2 Dataset

We use the PubMed abstracts dataset [1] in our experiments. This dataset contains articles published in bio-medical journals. We use the articles published between 2012 to 2015 for training topic models. This subset contains about 300,000 documents. Following [15], we generate 500 documents with a high value of diversity and 500 documents with a low value of diversity. We create high diversity documents as follows: we first randomly select 10 pairs of journals. Each pair contains two journals that are relatively unrelated to each other (we select 20 journals in total). For each pair of journals A and B, we select 50 articles to create 50 new probability distributions over topics as follows: we randomly select one article from A and one article from B and generate a document by averaging the selected articles' bag of topic counts. In this way, for each pair of journals we generate 50 documents with a high diversity value. We create low diversity documents as follows: for each of the chosen 20 journals, we perform a similar procedure but instead of choosing articles from two different journals, we select them from the same journal and generate 25 non-diverse documents. In the final set we have 500 diverse and 500 non-diverse documents.

3.5.3 Baselines

Our baseline for the topical diversity task is the method proposed in [15], which uses LDA for measuring topical diversity of documents. As an additional baseline, we use PTM [176] instead of LDA for measuring topical diversity. PTM is the state-of-the-art in topic modeling approaches, and based on our results PTM is more effective than the method proposed in [15]. Thus, PTM is our main baseline in this task.

3.5.4 Metrics

To measure the performance of topic models in the topical diversity task, we follow [15] and report ROC curves and AUC values. As another evaluation measure, we report the *sparsity* of topic models: the average number of topics assigned to the documents of a corpus [176]. This measure reflects the ability of topic models to achieving sparse $P(t\,|\,d)$ distributions. We also measure the *coherence* of the extracted topics. This measure indicates the purity of $P(w\,|\,t)$ distributions and a high value of coherence implies high purity within topics. For estimating the coherence of a topic model we use a reference corpus. As our reference corpus, we use a version of English Wikipedia.[2] We

[2] We use a dump of June 2, 2015, containing 15.6 million articles.

estimate the coherence of a topic model using normalized pointwise mutual information between the top N words within a topic using the following equation [119, 146]:

$$NPMI(T) = \sum_{t \in T} \sum_{w_i, w_j \in topN(t) \wedge i<j} \frac{\log \frac{P(w_i,w_j)}{P(w_i)P(w_j)}}{-\log(P(w_i, w_j))}, \tag{3.5}$$

where T is the set of extracted topics; $topN(t)$ is the top N most probable words within topic t; w_i is a word; $P(w_i, w_j)$ is estimated based on the number of documents in which w_i and w_j co-occur divided by the number of documents in the reference corpus; and $P(w_i)$ is estimated similarly, using maximum likelihood estimation.

3.5.5 Preprocessing

We first lowercase all the text in the corpus. Then, we remove the stopwords included in the standard stop word list from Python's NLTK package. In addition, we remove the 100 most frequent words in the collection and words with fewer than five occurrences.

3.5.6 Model parameters

As noted above, the topic modeling approach used in our experiments with HiTR is LDA. Following [15, 169, 176] we set the number of topics to 100. We set the two hyperparameters to $\alpha = 1/T$ and $\beta = 0.01$, where T is the number of topics, following [146]. In the re-estimation process, at each step of the EM algorithm, we set the threshold for removing unnecessary components from the model to 0.01 and remove terms with an estimated probability less than this threshold from the language models, as in [80].

We perform 10-fold cross validation, using 8 folds as training data, 1 fold as development set to tune the parameters, and 1 fold for testing.

3.5.7 Statistical significance

For statistical significance testing, we compare our methods to PTM using paired two-tailed t-tests with Bonferroni correction. To account for multiple testing, we consider an improvement significant if: $p \leqslant \alpha/m$, where m is the number of conducted comparisons and α is the desired significance. We set $\alpha = 0.05$. In Section 3.6, $^\blacktriangle$ and $^\blacktriangledown$ indicate that the corresponding method performs significantly better and worse than PTM, respectively.

3.6 Results

In this section, we first present the results of HiTR in the topical diversity task. Then, we analyze each individual level of re-estimation.

3.6.1 Topical diversity results

Figure 3.3 plots the performance of our topic models across different levels of re-estimation, and the models we compare to, on the PubMed dataset. We plot ROC curves and compute AUC values. To plot the ROC curves we use the diversity scores calculated for the generated pseudo-documents with diversity labels. HiTR improves the performance of LDA by 17% and PTM by 5% in terms of AUC. From Figure 3.3 two observations can be made.

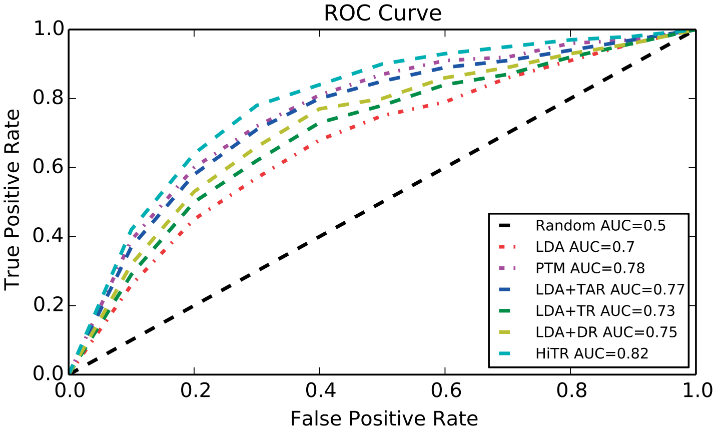


Figure 3.3: Performance of topic models in topical diversity task on the PubMed dataset. The improvement of HiTR over PTM is statistically significant (p-value<0.05) in terms of AUC.

First, HiTR benefits from the three re-estimation approaches it encapsulates by successfully improving the quality of estimated diversity scores. Second, the performance of LDA+TAR, which tries to address the generality problem, is higher than the performance of LDA+TR, which addresses impurity. General topics have a stronger negative effect on measuring topical diversity than impure topics. Also, LDA+DR outperforms LDA+TR. So, removing impurity from $P(t \mid d)$ distributions is the most effective approach in the topical diversity task, and removing impurity from $P(w \mid d)$ distributions is more effective than removing impurity from $P(w \mid t)$ distributions. Table 3.1 illustrates the difference between LDA and HiTR with the topics assigned by the two methods for a non-diverse document that is combined from two documents from the same journal, entitled “Molecular Neuroscience: Challenges Ahead” and “Reward Networks in the Brain as Captured by Connectivity Measures,” using the procedure described in Section 3.5.2. As only a very basic stopword list was applied, words like *also* and *one* still appear. We expect to have a low diversity value for the combined document. However, using Rao’s diversity measure, the topical diversity of this document based on the LDA topics is 0.97. This is due to the fact that there are three document-specific topics—topics 1, 2 and 4—and four general topics. Topics 1 and 2 are very similar and the δ between them is 0.13. The δ between, the other, more general topics is high; the average δ value between pairs of topics is as high as 0.38. For the same document, HiTR only assigns three document-specific topics and they are more

Table 3.1: Topic assignments for a non-diverse document using LDA and HiTR. Only topics with $P(t\,|\,d) > 0.05$ are shown.

LDA

Topic	$P(t\,|\,d)$	Top 5 words
1	0.21	brain, anterior, neurons, cortex, neuronal
2	0.14	channel, neuron, membrane, receptor, current
3	0.10	use, information, also, new, one
4	0.08	network, nodes, cluster, functional, node
5	0.08	using, method, used, image, algorithm
6	0.08	time, study, days, period, baseline
7	0.07	data, values, number, average, used

HiTR

Topic	$P(t\,|\,d)$	Top 5 words
1	0.68	brain, neuronal, neurons, neurological, nerve
2	0.23	channel, synaptic, neuron, receptor, membrane
3	0.09	network, nodes, cluster, community, interaction

pure and coherent. The average δ value between pairs of topics assigned by HiTR is 0.19. The diversity value of this document using HiTR is 0.16, which indicates that this document is non-diverse.

Next, Table 3.2 shows the sparsity of $P(t\,|\,d)$ using different topic models. All topic models that have TAR level of re-estimation achieve very sparse topic models. Thus, TAR contributes more to the sparsity achieved by HiTR. TAR increases the sparsity of LDA by more than 80%. This sparsity leads to improvements over the performance of LDA on the topical diversity task, which indicates that TAR is able to remove general topics from documents. Topic models achieved by PTM are slightly more sparse than those achieved by HiTR. However, the difference is not statistically significant. The fact that HiTR outperforms PTM indicates that PTM extremely parsimonizes documents and throws away essential information from documents while HiTR removes mostly non-essential information from documents.

3.6.2 HiTR results

In this section, we analyze different levels of re-estimation to get insights on how different levels on re-estimation work individually and how much they are successful in removing non-necessary information from documents, topics, and topic-assignments.

Document re-estimation results

In this section, we focus on answering **RQ2.1** What is the effect of DR on the quality of topic models? Can DR replace manual pre-processings?

DR outperforms LDA by 7% in measuring documents' topical diversity in terms of AUC. It also outperforms TR in this task but the difference is not significant. In fact,

Table 3.2: Sparsity of topic models trained on PubMed for the topical diversity task. For significance tests we consider p-value $< 0.05/7$.

Method	**Sparsity**
LDA	13.77
PTM	**1.78**
LDA+DR	13.17$^\blacktriangledown$
LDA+TR	12.35$^\blacktriangledown$
LDA+TAR	2.12
LDA+DR+TR	11.46$^\blacktriangledown$
LDA+DR+TAR	2.01
LDA+TR+TAR	1.92
HiTR	1.80

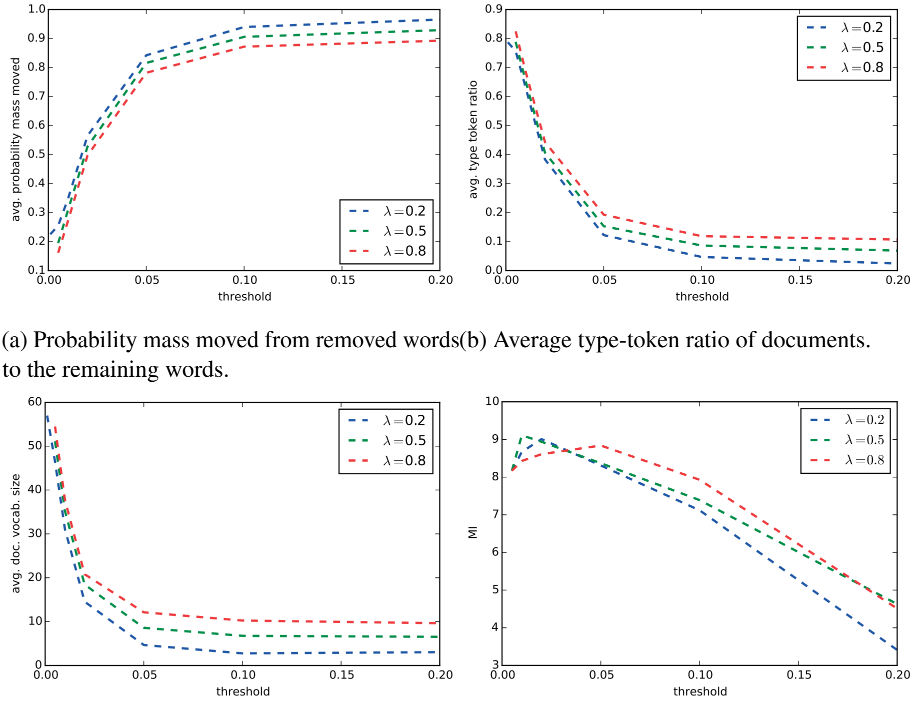
Table 3.3: The effect of document pre-processing on the quality of topic models measured in terms of coherence and AUC achieved in the topical diversity task.

Method	$Coherence$	AUC
LDA (without pre-processing)	6.23	0.54
LDA+pre-processing	8.45	0.73
LDA+DR	8.95	0.75
LDA+DR+TR	10.29	0.79

DR and TR are addressing the same problem with topic models. Both are successful in addressing *impure topics*. However they are not successful in addressing the *general topics* problem, since they have a high value of sparsity.

To analyze the effectiveness of DR in re-estimating documents and addressing the problems with topic models, we design an experiment in which no manual pre-processing is done and topic models are trained on these not-pre-processed documents. Our expectation is that even without doing any pre-processing a method that addresses the generality problems with topic models should still be able to achieve a good performance and do the pre-processing implicitly and automatically. Since DR tries to pre-process documents automatically, it should achieve a high quality topic model on these datasets. Table 3.3 shows the performance of LDA, DR, and LDA+DR+TR in terms of their coherence. As expected, the coherence of LDA decreases by more than 23% when no pre-processing is done on documents. More interestingly, adding DR scores better, both in terms of coherence and AUC, than manual pre-processing.

Next, we analyze the effect of the amount of document re-estimation on the quality of topic models. We control the amount of re-estimation by the values of the parameters of DR: λ and $threshold$. Figure 3.4 shows the effect of different values of the parameters on documents and its impact on the quality of trained topic models. Two conclusions can be drawn. First, λ does not have a great impact on the documents' statistics as even with very different values of λ documents have similar statistics. The threshold has a bigger impact on the documents. Second, although the statistics of documents are similar for different values of λ, the thresholds for which the best coherence is achieved for them,



(a) Probability mass moved from removed words to the remaining words.

(b) Average type-token ratio of documents.

(c) Average document vocabulary size.

(d) Coherence of topic models estimated using Equation 3.5.

Figure 3.4: The effect of different values of the parameters of DR on the documents in terms of their probability mass moved, type-token ratio, and vocabulary size and its effect on the quality of trained topic models in terms of their coherence.

are very different. For $\lambda = 0.5$ the best coherence is achieved for $threshold = 0.01$, while for $\lambda = 0.8$ the best coherence is achieved for $threshold = 0.05$. This indicates that there is a correlation between these parameters. As expected, when λ is high, which corresponds to less re-estimation, the threshold should be high to remove unnecessary words from documents.

Topic re-estimation results

To answer **RQ2.2**, we now focus on the TR level of HiTR. Since TR tries to remove the impurity from topics, we expect TR to increase the coherence of the topics by removing unnecessary words from topics. Table 3.4 shows the top five words for some example topics calculated from the PubMed dataset, before and after applying TR. These examples indicate that TR can successfully remove general words from topics.

We measure the purity of topics based on the coherence of words within $P(w \mid t)$ distributions. Table 3.5 shows the coherence of topics according to different topic modeling approaches, in terms of average mutual information. More coherent topics are beneficial, because they are an indicator of more pure topics, which are essential to

Table 3.4: Examples of topics before and after applying topic re-estimation on the PubMed dataset.

	Before TR		After TR	
Topic t	w	$p(w\,\vert\,t)$	w	$p(w\,\vert\,t)$
1	women	0.07	women	0.06
	men	0.02	men	0.05
	costs	0.02	health	0.05
	per	0.02	costs	0.03
	total	0.02	economic	0.02
2	using	0.01	algorithm	0.04
	method	0.01	method	0.03
	used	0.01	data	0.03
	algorithm	0.01	performance	0.02
	data	0.01	system	0.01
3	sequences	0.02	genome	0.05
	genome	0.02	sequences	0.04
	genes	0.02	genes	0.03
	using	0.01	genomic	0.03
	two	0.01	gene	0.02

Table 3.5: The coherence of different topic models in terms of average mutual information between top 10 words in the topics calculated using Equation 3.5 on the PubMed dataset.

Method	**Coherence**
LDA	8.17
PTM	9.89
LDA+TR	9.46
LDA+DR+TR	10.29$^{\blacktriangle}$

achieving a good performance in topical diversity task. TR increases the coherence of topics by removing the impure parts from topics. The coherence of PTM is higher than the coherence of TR. However, when we first apply DR, train LDA, and finally apply TR, the coherence of the extracted topics is significantly higher than the coherence of topics extracted by PTM. From these findings we conclude that TR is effective in removing impurity from topics. Moreover, DR also contributes in making topics more pure.

To see how much impurity is being removed from topics by using TR, we investigate the effect of TR on the distribution of words within topics and we measure the number of words and the re-allocated probability mass within topics before and after TR. Figure 3.5 shows the probability mass of the words left after TP is applied to the topics of the original LDA model. The average number of words within extracted topics from the PubMed dataset is about 337 without TR, and about 181 after performing TR. On average, the words that are not removed by TR take 41% of the probability mass in the

LDA topic models (the dotted red line in Figure 3.5). In the re-estimated topic model, they occupy the full 100% of the probability mass. Thus, after applying TR, the topic models become more sparse, and the remaining topic-specific words receive higher probabilities. As shown in the figure, over all topics, after applying TR, the probability mass is re-allocated and some words are removed.

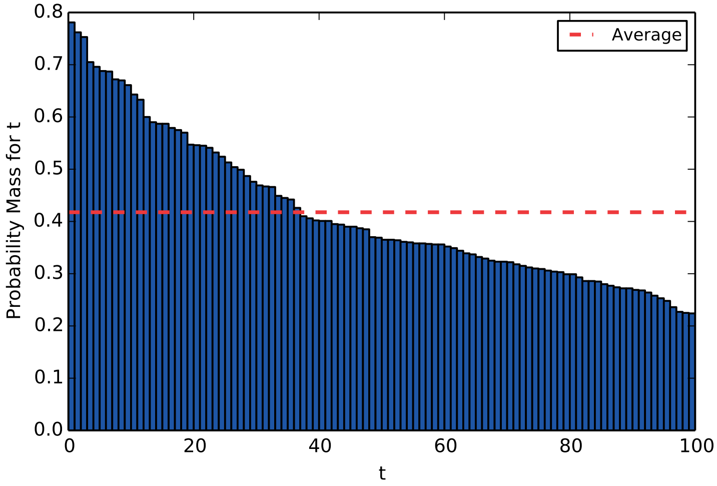


Figure 3.5: Probability mass of the words left after TR in the topics of the original LDA model. The y-axis shows $\sum_{\{w \mid P_{LDA+TR}(w \mid t)>0\}} P_{LDA}(w \mid t)$ for a topic t.

Topic assignment re-estimation results

To answer **RQ2.3**, we now turn to the TAR level of HiTR. We are interested in seeing how HiTR deals with the issue of general topics. General topics are topics that, for many documents, have a high probability of being assigned. To gain insight in how LDA and HiTR perform in this respect, we sum the probability of assigning a topic to a document, over all documents: for each topic t, we calculate $\sum_{d \in C} P(t \mid d)$, where C is the collection of all documents. Figure 3.6 shows the distribution of probability mass before and after applying TAR. General topics naturally have high values as they are assigned to most of the documents with high probability. In Figure 3.6 the topics are sorted based on the topic assignment probability of LDA. As we can see from Figure 3.6, LDA assigns a vast portion of the probability mass to a relatively small number of topics. These topics are mostly general topics that are assigned to most of documents. We expect, however, that many topics are represented in some documents, while relatively few topics will be relevant to all documents. When TAR is applied, the distribution is less skewed and the probability mass is more evenly distributed.

There are some topics that have a high $\sum_d P(t \mid d)$ value in LDA's topic assignments and a high $\sum_d P(t \mid d)$ value after applying TAR as well (they are marked as "non-general topics" in Figure 3.6). Table 3.6 shows the top five words for these topics. Although these topics contain some general words such as "used," they are not general topics.

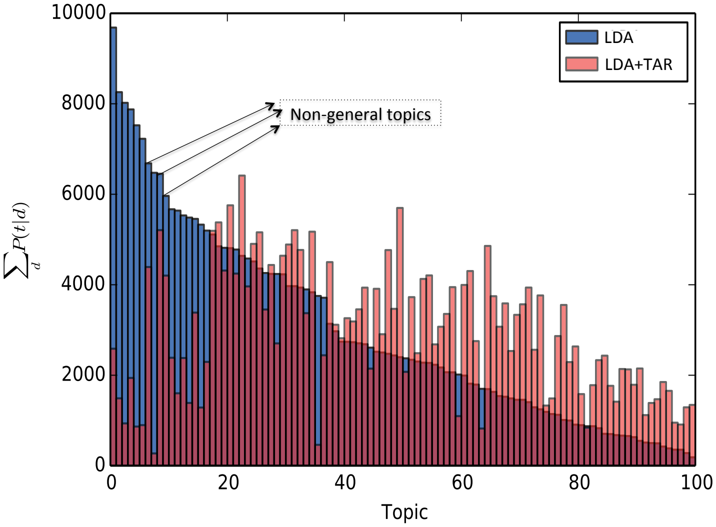


Figure 3.6: The total probability of assigning topics to the documents in the PubMed dataset estimated using LDA and LDA+TAR. (The two areas are both equal to the number of documents ($N \approx 300K$)).

TAR is able to find these three non-general topics and their assignment probabilities to documents in the $P(t\,|\,d)$ distributions is not changed as much as the actual general topics.

Table 3.6: Top five words for the topics marked as "non-general topics" in Figure 3.6.

Topic	Top 5 words
1	health, services, public, countries, data
2	surgery, surgical, postoperative, patient, performed
3	cells, cell, treatment, experiments, used

To further investigate whether TAR really removes general topics, in Table 3.7 we show the top five words for the first 10 topics in Figure 3.6, excluding the topics marked as "non-general topics" in the figure. These seven topics have the highest decrease in $\sum_d P(t\,|\,d)$ values when we apply TAR. As can be seen from Table 3.7, the topics contain general words and are not informative. In the figure, we can see that after applying TAR, the $\sum_d P(t\,|\,d)$ values are decreased dramatically for these topics and that the mass is re-distributed across other topics, without creating new general topics that apply to nearly all documents. We can conclude that TAR can correctly distinguish general from specific topics and re-assign probability mass accordingly.

Table 3.7: Top five words for the topics detected by TAR as general topics.

Topic	Top 5 words
1	use, information, also, new, one
2	ci, study, analysis, data, variables
3	time, study, days, period, baseline
4	group, control, significantly, compared, groups
5	study, group, subject, groups, significant
6	may, also, effects, however, would
7	data, values, number, average, used

3.6.3 Parameter analysis

In this section, we analyze the effect of the λ parameter on the performance of DR, TR, and TAR in the topical diversity task. Figure 3.7 displays the performance at different levels of re-estimation based on a range of values for λ. Recall that with $\lambda = 1$, no re-estimation takes place, and all methods equal LDA. The following interesting observations can be made from this figure.

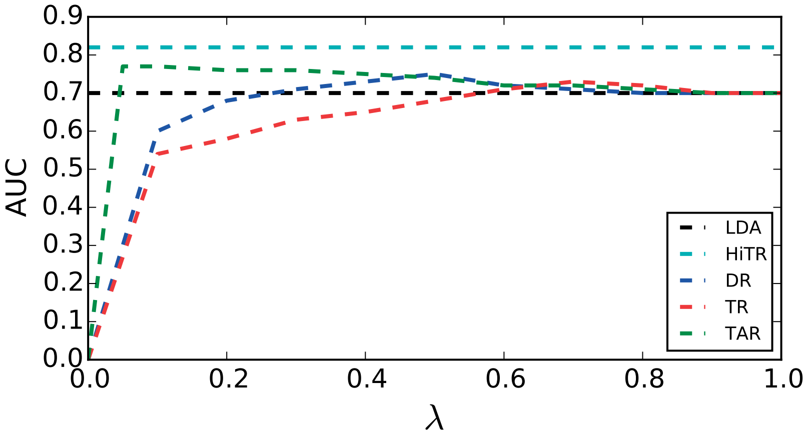


Figure 3.7: The effect of the λ parameter on the performance of topics models in topical diversity task on PubMed dataset.

First, DR reaches its best performance with moderate values of λ ($0.4 \leqslant \lambda \leqslant 0.45$). This reflects the fact that documents contain a moderate amount of general information and that DR is able to successfully deal with it. For $\lambda \geqslant 0.8$ the performance of DR and LDA is the same and for these values of λ DR does not increase the quality of LDA.

Second, the best performance of TR is achieved with high values of λ ($0.65 \leqslant \lambda \leqslant 0.75$). This indicates that topics usually only need a small amount of re-estimation. With this slight re-estimation, TR is able to improve the quality of LDA. However, for the values of $\lambda \geqslant 0.75$ the accuracy of TR degrades.

Third, TAR achieves its best performance with very low values of λ ($0.02 \leqslant \lambda \leqslant$

0.05). These low values of λ correspond to more re-estimation. From this result, we conclude that most of the noise is in the $P(t \mid d)$ distributions, and that aggressive re-estimation allows TAR to remove most of this noise. The best values of λ optimized for HiTR using the development set are close to the best values of λ according to Figure 3.7.

3.6.4 Impact of underlying topic model on the performance of HiTR

In this section, we analyze the effect of using PTM as the underlying topic model for HiTR on the performance of HiTR. We apply HiTR on top of PTM and compare the results with the results of applying HiTR on top of LDA. Table 3.8 shows the results of this experiment. The results show that: (1) Applying HiTR on top of PTM does not improve PTM's performance significantly. We believe, the reason is that PTM already removes a lot of general information from topics/documents, but in some cases it also removes non-general information. LDA is in the other side of the spectrum, it keeps all information (general and non-general), and HiTR removes general information and keeps only the non-general information which leads to a higher performance. (2) PTM benefits the most from the DR step. It shows that PTM is already effective in removing generality/impurity from topic-word and document-topic distributions, however it does not have a mechanism to remove generality/impurity from document-word distributions. (3) The performance of HiTR with LDA is significantly better than the performance of PTM and PTM with HiTR. As we mentioned, this shows that HiTR is more effective when the underlying topic model contains all information (general and non-general) and it can remove the non-general part. (4) In terms of sparsity, HiTR makes PTM more sparse, however the difference is not significant. Thus, applying HiTR on an already sparse topic model does not have a big influence on its sparsity.

Table 3.8: The performance and sparsity of HiTR using PTM as the underlying topic model in the topical diversity task.

Method	**AUC**	**Sparsity**
PTM	0.78	1.78
PTM+DR	0.79	1.73
PTM+TR	0.77	1.71
PTM+TAR	0.78	1.65
PTM+HiTR	0.79	1.63

3.7 Analysis

In this section, we want to gain additional insights into HiTR and its effects on topic estimation. Purity of topic assignments to documents based on $P(t \mid d)$ distributions has the highest effect on the quality of estimated diversity scores for documents. Therefore, it is important to measure how pure the estimated topic assignments are using HiTR. In this section, we measure how much impurity is removed by HiTR from topic distributions.

Then, we analyze the efficiency of HiTR.

Based on the topics assigned by HiTR, LDA and PTM, we perform document clustering and document classification. For clustering, following [146], we consider each topic as a cluster. Each document d is assigned to the topic that has the highest probability value in $P(t\,|\,d)$. For classification, we use all topics assigned to the document and consider them as features for a supervised classification algorithm. As the classification algorithm we use SVM. High accuracy achieved in document classification is then an indicator of high purity of topic distributions.

We note that our focus in this section is not on achieving a top performance in document clustering and classification tasks: we only consider these tasks as a means to assess the purity of topic distributions using different topic models.

3.7.1 Datasets

We use three datasets: 20-NewsGroups,[3] Reuters [124] and Ohsumed.[4] The Reuters dataset contains 806,791 documents with category labels for 126 categories. For clustering and classification of documents, we use the 55 categories in the second level of the category hierarchy. 20-NewsGroups contains 20 categories and around 1,000 documents in each category, so in total there are about 20,000 documents. The Ohsumed dataset contains 50,216 documents grouped into 23 categories.

3.7.2 Purity metrics

For measuring the purity of clusters, two standard evaluation metrics are used: *purity* and *normalized mutual information* (NMI) [131].

3.7.3 Settings

We evaluate document clustering and classification using 10-fold cross validation and perform the same document pre-processing as described in Section 3.5.5.

3.7.4 Purity results

Table 3.9 shows the purity of HiTR in the document clustering task. For all 3 datasets, on both measures, the purity of topics created by HiTR is significantly higher than with PTM. As expected, TAR is mostly responsible for the purity of $P(t\,|\,d)$: all runs which include TAR either improve or do not differ significantly from PTM. The different combinations show that also DR and TR yield additional purity, indicating that each of the three address different issues and contribute in a different way.

Table 3.10 shows the performance of different topic models on the document classification task. Again HiTR significantly outperforms PTM on all three datasets. We see the same trend as with clustering, but amplified: here all runs without TAR perform significantly worse than PTM. Note that on the smallest dataset, LDA and PTM performs already well, and so are harder to improve. Where in document clustering

[3] Available at `http://www.ai.mit.edu/people/~jrennie/20Newsgroups/`

[4] Available at `http://disi.unitn.it/moschitti/corpora.htm`

Table 3.9: Purity of topic models estimated in terms of purity achieved in document clustering. For significance tests, we consider p-value < 0.05/7.

	Reuters (N=806,791, C=55)		**20-Newsgroups (18,846, C=20)**		**Ohsumed (N=50,216, C=23)**	
Method	Purity	NMI	Purity	NMI	Purity	NMI
LDA	0.55	0.40	0.52	0.36	0.50	0.30
PTM	0.61	0.43	0.57	0.38	0.55	0.33
LDA+DR	$0.57^{\blacktriangledown}$	$0.41^{\blacktriangledown}$	0.56	0.39	$0.53^{\blacktriangledown}$	$0.32^{\blacktriangledown}$
LDA+TR	$0.57^{\blacktriangledown}$	$0.42^{\blacktriangledown}$	0.56	0.38	$0.53^{\blacktriangledown}$	$0.31^{\blacktriangledown}$
LDA+TAR	0.60	0.43	0.57	0.39	0.54	0.33
LDA+DR+TR	0.58	$0.42^{\blacktriangledown}$	0.57	0.38	0.54	0.32
LDA+DR+TAR	0.60	0.43	0.58	0.40	0.55	$0.35^{\blacktriangle}$
LDA+TR+TAR	0.61	0.43	0.58	$0.40^{\blacktriangle}$	$0.56^{\blacktriangle}$	$0.34^{\blacktriangle}$
HiTR	**0.64**$^{\blacktriangle}$	**0.45**$^{\blacktriangle}$	**0.60**$^{\blacktriangle}$	**0.42**$^{\blacktriangle}$	**0.57**$^{\blacktriangle}$	**0.35**

Table 3.10: Purity of topic models estimated in terms of accuracy achieved in document classification. For significance tests, we consider p-value < 0.05/7.

	Reuters (N=806,791, C=55)		**20-Newsgroups (N=18,846, C=20)**		**Ohsumed (N=50,216, C=23)**	
Method	Accuracy	Imp. over LDA	Accuracy	Imp. over LDA	Accuracy	Imp. over LDA
LDA	0.76	–	0.81	–	0.50	–
PTM	0.82	8%	0.87	7%	0.56	12%
LDA+DR	$0.79^{\blacktriangledown}$	4%	$0.83^{\blacktriangledown}$	2%	$0.52^{\blacktriangledown}$	4%
LDA+TR	$0.78^{\blacktriangledown}$	3%	$0.83^{\blacktriangledown}$	2%	$0.53^{\blacktriangledown}$	1%
LDA+TAR	0.82	8%	$0.85^{\blacktriangledown}$	5%	0.54	8%
LDA+DR+TR	$0.80^{\blacktriangledown}$	5%	$0.84^{\blacktriangledown}$	4%	$0.53^{\blacktriangledown}$	6%
LDA+DR+TAR	0.83	9%	0.86	6%	0.56	12%
LDA+TR+TAR	$0.82^{\blacktriangle}$	8%	0.87	7%	$0.58^{\blacktriangle}$	16%
HiTR	**0.85**$^{\blacktriangle}$	12%	**0.89**$^{\blacktriangle}$	10%	**0.60**$^{\blacktriangle}$	20%

only the topics with the highest probability are considered, in document classification the classifiers use the entire $P(t\,|\,d)$ distributions to classify documents. Performance of all methods in document classification is closer to the perfect classifier than their performance in document clustering, as the maximum value of both accuracy and purity is 1. This indicates that the most probable topic does not necessarily contain all information about the content of a document. In the cases that a document is about more than one topic, the classifier utilizes all $P(t\,|\,d)$ information and performs better. Therefore, the higher accuracy of HiTR in this task is an indicator of its ability to assigning document-specific topics to documents.

3.7.5 HiTR's efficiency

Table 3.11 shows the execution times of HiTR, LDA, and PTM. The reported execution time for HiTR is the time it took to run HiTR once, given the corpus as input and topic assignments to documents as output. All models were run on machines with 6-core 3.0 GHz processors. The results show that, even on large datasets, HiTR does not add much complexity to LDA and the difference between the execution times of LDA and HiTR are reasonable. The execution times of PTM grow much faster than those of LDA and HiTR when the number of documents increase.

Table 3.11: The execution time of HiTR, LDA, and PTM in hours. N and $\#w$ are the number of documents and tokens in the corpus, respectively.

Dataset	**Method**	**Hours**
Reuters	LDA	6.18
$N = 807K$	PTM	26.00
$\#w = 1,5M$	HiTR	9.17
20-NewsGroups	LDA	1.13
$N = 19K$	PTM	0.93
$\#w = 5,2M$	HiTR	1.45
Ohsumed	LDA	1.42
$N = 50K$	PTM	3.88
$\#w = 10M$	HiTR	2.45

3.8 Conclusion

We have proposed Hierarchical Topic model Re-estimation (HiTR), an approach for re-estimating topic models and applied them to measure topical diversity of text documents.

We have shown by experimental means that our approaches are able to remove general topics from topic models and increase the purity of topics. The results show that the estimated diversity scores for documents using HiTR are more accurate than those extracted using topic models created by LDA and PTM. Our three main findings are as follows. First, general topics have the largest negative impact on the quality of topic models when they are used for measuring topical diversity. This indicates that purity of topic assignments is more important than purity of the distribution of words in topics and the distribution of words in documents in topical diversity task. The topic assignment re-estimation (TAR) that is designed to address this problem successfully detects general topics and removes them from documents. Second, re-estimation at each level helps to improve the quality of estimated diversity scores. We have shown that these “cleaned document topic models” yield better results when applied to measure topical diversity of documents. However, to achieve a highly accurate diversity scores, re-estimation at all three levels is needed to improve on the state-of-the-art PTM approach. Third, we analyzed the effectiveness of HiTR in two other tasks: document clustering and document classification. We found that HiTR can achieve higher performances in these tasks compared to LDA and PTM. This finding suggest that although HiTR is originally designed for better estimation of topical diversity, it can be applied in a wider variety of tasks.

Our proposed approach has some limitations. First, HiTR is most effective at removing general information from the three probability distributions mentioned. However, to train a more accurate topic model which has a good performance on the topical diversity task it is also important to remove very specific words from documents. Current approaches, including HiTR, are not able to address this problem adequately. Second, the experiments on the topical diversity task are conducted in an artificially created dataset. More robust datasets are needed for evaluating HiTR in this task.

There are several future directions. In principle, HiTR is a re-estimation method that can be applied to any topic model to enhance its quality. In this study, we have applied HiTR to LDA and PTM. In our future work, we plan to examine the effect of HiTR on a wide range of topic models besides LDA and PTM such as PLSA. In this research, we adapted and used Rao's diversity measure for estimating diversity of documents. There are several other diversity measures proposed in biology such as Functional Divergence and Functional Attribute Diversity.

In this chapter, we studied how we can measure the topical diversity of documents and use this measurement for managing documents. Next, in Chapter 4, we turn our attention towards the email domain and study how we can help users manage their emails more efficiently.

4
Domain Adaptation for Commitment Detection in Email

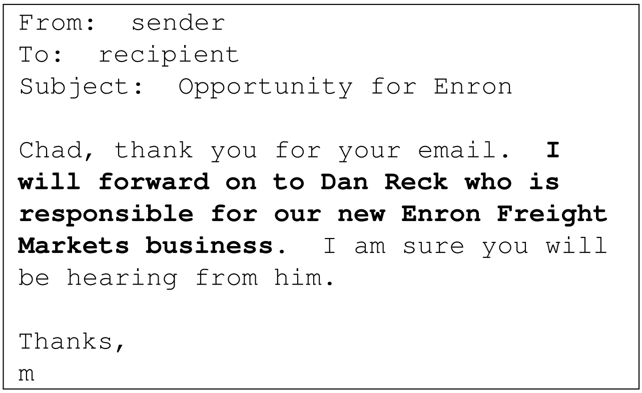
In this chapter, we focus on automatic task management systems and study how we can detect commitments made in email to help users manage their ongoing tasks more efficiently. Our aim is to answer **RQ3**.

4.1 Introduction

Email is an important communication medium for individuals and organizations. People use email not only as a communication tool, but also as a means to create and manage tasks [20, 43, 145, 189]. When the number of ongoing tasks created via emails increases, people can struggle to manage their tasks and monitor their progress [17, 207]. Automatic task management systems can overcome this problem and help people manage their tasks more efficiently [17, 74]. Commitments such as "I'll send the report by the end of day" are one type of task that involve promises made between individuals to complete future actions [41]. Figure 4.1 shows an example commitment in an email. Such tasks are often hidden in emails, and users can struggle to recall and complete them in a timely manner. Detecting commitments automatically enables task management tools and digital assistants to generate reminders and notifications to help users meet their obligations. Despite the potential benefits of automatic commitment detection, work in this important area has been limited to only a handful of studies [41, 50, 115, 116, 128].

Commitment detection is challenging for at least two reasons. First, the commitment detection task itself is inherently difficult, even for humans [115], and particularly difficult when the text is short, with limited context. Second, email models are trained and shipped to users based on potentially biased datasets. For both privacy and practical reasons, email-based models are often trained on public datasets, which are skewed in a variety of ways. For example, Enron [107] and Avocado [147], two commonly-used email datasets for learning email models, belong to two organizations with different focus areas and from different time periods. Terminology, including named entities and technical jargon, can vary greatly across domains and over time. As such, models learned on one dataset may be biased and might not perform well on a different target

This chapter was published as [13].



From: sender
To: recipient
Subject: Opportunity for Enron

Chad, thank you for your email. **I will forward on to Dan Reck who is responsible for our new Enron Freight Markets business.** I am sure you will be hearing from him.

Thanks,
m

Figure 4.1: A sample email from the Enron email corpus with a commitment sentence highlighted. The commitment detection task is to automatically detect such sentences in email.

dataset. Biases will affect all models trained on email corpora. Transfer learning [150] is a domain adaptation method that enables transferring knowledge learned in one domain (source domain) to another domain (target domain). It has been shown that this approach is successful for addressing domain differences in many tasks including text and image classification [161, 212], sentiment analysis [26], and collaborative filtering [125]. Using this approach for transferring knowledge learned from one email collection to another may help achieve more robust and generalizable models for commitment detection.

In this study, we evaluate the efficacy of transfer learning for commitment detection. To learn commitment detection models, we first create two datasets with commitments occurring in emails sourced from the Enron and Avocado collections. To do this, we extract sentences from emails and ask trained annotators to assign binary commitment labels to the sentences. We argue that commitment vocabulary is mostly domain independent and that a transfer learning approach can help remove domain-specific information from commitment models and capture the core language of commitments.

To learn a domain-independent commitment model, we first try to characterize the differences between domains (email corpora) and then use this characterization to transfer knowledge between them. We investigate the performance of domain adaptation methods working at different granularities: feature-level adaptation and sample-level adaptation. For feature level adaptation, we first learn a mapping between features (n-grams) of source and target domains and use the mapping to transfer the features of the source domain to the target domain and learn the classifier using the transferred data. For sample-level adaptation, we use importance sampling (IS) [172] to weight training samples in the source domain based on their similarity to the samples of the target domain. We use the weighted samples in the source domain to learn the classifier and apply it to the samples in the target domain. We further combine sample-level and feature-level domain adaptation using a neural autoencoder [19]. The autoencoder is trained to maximize the commitment classification accuracy while minimizing the reconstruction loss. Moreover, to remove domain specific information from representations of samples, a domain classifier is added to the autoencoder. Given a training sample

from the source/target domain, the domain classifier attempts to predict the domain to which the sample belongs. During training the accuracy of the domain classifier is minimized. By minimizing domain classifier accuracy, domain-specific information is removed from the samples.
In this chapter we address our third research questions, **RQ3**:

> *Can commitments be reliably detected in emails? And how does the performance of commitment models change when they are tested on a different domain than they are trained on?*

Our main research contributions are as follows:

- Analysis of the impact of domain transfer on commitment detection. We show that the quality of commitment detection degrades significantly as we apply commitment detection models across domains.
- Proposing different approaches for characterizing differences between email corpora and show that domain adaptation (specifically, transfer learning) can remove domain-specific bias from commitment detection models.
- Demonstrating through extensive analysis that domain adaptation methods lead to significant gains in the precision and recall of commitment detection models.

4.2 Related Work

Prior research in a number of areas applies to the study presented in this chapter. This includes work on automatic email management [17, 20, 41, 128, 207] and task progress monitoring [17, 43, 74]. Prior research on commitment detection and domain adaptation is particularly relevant, and we describe it in detail in this section.

4.2.1 Commitment detection

The detection of commitments in email has been the subject of several prior studies. Lampert et al. [114, 115] show that the annotation of commitments and requests in email is challenging, even for humans. They devise guidelines for collecting judgments and building datasets for commitment/request classification. The most interesting insight is that when statements are given in context (full email) the annotation task is easier and more accurate. Although the findings of these studies can help design commitment detectors, no automatic commitment classifiers are trained in these studies. De Felice [50] proposes a fine-grained classification of commitments in emails. She further studies which phrases are associated with commitments. As with [114, 115], no automatic commitment detector is created. Automatic commitment classifiers have been developed in prior work. Cohen et al. [41] train classifiers to classify sentences in email into one of the following speech acts: deliver, commit, request, amend, propose, meeting. They represent sentences by TF-IDF weighted vectors over word n-grams. Kalia et al. [102] use more sophisticated features for detecting commitments in email such as named entities, part-of-speech (POS) tags, dependencies, and co-reference resolution. They consider both requests and commitments as commitments. They run experiments on

Enron and an instant messaging dataset (Hewlett-Packard's IT incident management chat logs). Corston-Oliver et al. [43] use different kinds of features for detecting tasks in email. They consider commitments as one of the tasks they try to extract from email. Lampert et al. [116] also consider both requests and commitments. They use a set of features such as message length, the presence of modal verbs, and question words, and train an classifier on a set of manually-labeled emails. Their main conclusion is that only some regions of emails are relevant to commitment detection, and other regions often introduce noise.

All these studies use a single dataset for training models and performing analysis. The datasets used in these studies are also small. The studies do not consider the important challenge of domain bias or domain adaptation. Although Cohen et al. [41] performed a limited analysis on the transferability of commitment models, they used datasets from the same domain.

4.2.2 Domain adaptation

Domain adaptation is the ability to learn a model from data in a source domain and adapt it to have a good performance on a different target domain [18]. Domain adaptation methods can be grouped into two categories: sample-level adaptation and feature-level adaptation methods [150, 234]. Sample level adaptation methods attempt to remove domain bias by weighting the samples in a way that the difference between the distribution of the weighted samples in the source domain and the distribution of samples in target domain is reduced. The most common sample weighting approach is importance sampling, in which source samples are re-weighted based on their similarity to samples in the target side [172]. TrAdaBoost [45] exploits a similar idea, but the re-weighting is performed iteratively in a boosting fashion. However, unlike the importance sampling method, TrAdaBoost needs labeled samples in the target domain. Similar ideas have been used in other studies for sample-level domain adaptation [94, 233].

Feature-level adaptation techniques try to remove the domain bias from features by removing domain specific features or transforming them from source to target domain [109, 150, 234]. Learning a mapping between features of different domains have been studied in machine translation [117, 137]. Here the domains are two different languages. With the success of deep autoencoders for learning unsupervised feature representations, these models have been used for domain adaptation [36, 73, 234]. The main intuition behind these methods is using a set of combined samples from source and target domains to learn a representation that is domain independent. After creating the representation, a classifier can be trained on the de-biased representations. Zhuang et al. [234] integrate the representation learning step of the autoencoders with the classifier learning step. They use an autoencoder that tries to learn a representation that is both domain independent and leads to a good performance in classification of samples in the target domain. In this study, we apply a similar idea, however, we try to further remove the domain bias by introducing a domain classifier. Moreover, Zhuang et al. [234] adapts the model for image classification and it is not straightforward to use this model for detecting commitments.

Recently, adversarial training has been applied to domain adaptation [28, 37, 69, 70, 130]. The main intuition behind these methods is adding a domain classification loss to the task's loss and trying to maximize the domain loss. Maximizing the domain loss ensures that the learned representations do not contain any domain information. We also use a similar approach to adversarial training [70] for learning representations of sentences. We further extend this work by using a sequence-to-sequence autoencoder [182] to the model for learning representations. The main reason for having a sequence to sequence autoencoder is that this model is a powerful means for encoding text. Having this autoencoder in the pipeline means that we can learn accurate sentence representations that capture the information present in word sequences.

4.3 Task and Data

In this section, we introduce the commitment detection task and the dataset we collected for training commitment models.

4.3.1 Detecting commitments in email

As in [43], we define a commitment as any sentence in an email where the sender is promising to do an action which can potentially be added to his/her TODO list (e.g., sending a document) or be worthy of a reminder (e.g., meeting a colleague). We model commitment detection as a binary classification task.

More precisely, the input in the commitment detection task is a sentence in an email and the output is a binary label indicating whether that sentence constitutes a commitment between the sender and the recipient of the email. We assume that there is a set of sentence-judgement pairs $X = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the sentence and y_i is the binary commitment label for x_i assigned by annotators. We use X to train a model and then use it to predict commitment labels for new sentences. We use three different representations to represent sentences in X: (1) bag of word n-grams: for feature based models (such as logistic regression (LR)) we extract word n-grams (we set $n = 3$) and represent the sentences by the frequency of word n-grams in them. (2) bag of part-of-speech (POS) n-grams: similar to the previous representation but with bag of POS-tagged word n-grams (we set $n = 3$) in sentences instead of bag of word n-grams. We use SPLAT [160] for extracting POS tags. (3) sequence of words: for sequence-based models (such as autoencoders), we represent sentences as sequences of words.

4.3.2 Collected dataset

We use the Enron [107] and the Avocado [147] datasets to construct commitment datasets. In this section, we briefly describe these datasets, the crowd-sourcing task where third-party annotators labeled sentences for whether they constituted commitments, and the collected commitment datasets.

Crowd-sourcing

The Enron dataset contains emails from 158 users who were mostly senior management of Enron, a natural gas transmission corporation. There are about 200K emails in this dataset. The Avocado dataset contains about 940K emails from employees of a defunct information technology company.

To construct crowd-sourcing tasks, we first extract sentences from emails. From the Enron dataset, we randomly select 61,398 sentences and ask annotators to assign commitment labels to them. A random sample of sentences does not contain a high number of positive samples. Therefore, to collect more positive samples, we train an LR classifier on the collected dataset and use it to extract sentences that are more likely positive. We use bag of word n-grams representations to train the LR model. We first use the trained classifier to assign to each sentence the probability of belonging to the positive class. Then, we perform a weighted random sampling based on these probabilities and select an additional 4,000 sentences to annotate. For extracting sentences from Avocado dataset we use the trained LR model to weight sentences and then we select 13,021 sentences for annotation.

After extracting sentences from Enron and Avocado datasets, we ask crowd-workers to assign commitment labels to the sentences. Each annotation task contains a sentence highlighted in an email and the following question: “Does the highlighted sentence contain a specific action that the sender must complete or is obliged to do? (The action must be on the sender and must not already be complete).” If the answer for this question is yes, we consider it a commitment. Each sentence is labeled by two annotators. If there is a disagreement between two annotators, then the sentence is annotated by a third annotator. A sentence is considered positive if at least two annotators annotate it as positive. The inter-annotator agreement between the annotators based on Krippendorff’s α is 0.73, indicating a substantial agreement between annotators [3].

Commitment datasets

Table 4.1 shows the statistics of the created datasets. We only use the annotated sentences as samples and ignore the rest of the email. Enron is a much larger dataset than Avocado. Since most of the samples are picked randomly from the dataset, it contains many more negative samples than positive ones. Conversely, since Avocado samples are picked based on the outputs of a machine-learned classifier, it contains more positive samples. Therefore, this dataset is biased toward the classifier and the distribution of positive and negative samples in this dataset does not reflect the true base rate.

Table 4.1: The statistics of the commitment datasets.

	Enron	**Avocado**
# samples	65,398	13,021
# positive samples	3,337	4,484
avg. sentence length	12.1	14.5
median sentence length	10	13

Table 4.2 shows top 10 most informative Enron n-grams for the positive class extracted based on pointwise mutual information. The Jaccard similarity of this set with the top 10 positive class Avocado features is 43%. (Due to licensing restrictions, the

Avocado features may not be published.)

Table 4.2: The most informative Enron features associated with the positive class.

“i will”, “i”, “will”, “i’ll”, “let you know”, “let you”, “call you”, “i shall”, “we will”, “will call”

4.4 Transfer Learning for Detecting Commitments

In this section, we describe the approaches used in this study for transferring knowledge between email datasets for the task of commitment detection. Given a set of labeled samples in the source domain S, our goal is to create a model that has a high commitment detection accuracy when it is applied in the target domain T. We use three different approaches for transferring classification knowledge between email domains: feature-level adaptation, sample-level adaptation, and an autoencoder that attempts to leverage both feature- and sample-level adaptation.

4.4.1 Feature-level adaptation

In this section, we introduce our approaches for adapting feature-level domain information. We use two feature-level adaptation techniques: feature selection and feature mapping.

Feature selection

The main intuition of feature selection approach for domain adaptation is detecting domain specific features in source and target domains and removing them from the train and test samples. We assume that we have a set of unlabeled samples, D_S, in the source domain and a set of unlabeled samples, D_T, in the target domain(s). To remove domain specific features from S, we first train a domain classifier using samples in D_S as positives and samples in D_T as negatives. The classifier is trained to discriminate samples of D_S from samples of D_T. Therefore, the most discriminative (informative) features of the classifier are considered domain-specific features. We use logistic regression (LR) for training the classifier and select top K features from the classification model (we set $K = 1000$) and finally replace the selected features with a unique symbol (‘DOMAINWORD’) in training samples from the source domain. We follow a similar procedure to remove domain-specific words from samples of the target domain. After removing domain specific information from the source and target domain samples, we train a commitment classifier on samples from the source domain and directly use that to predict the commitment labels in the target domain. A sample of features that are strongly associated with the Enron domain (the most informative features regarding the Enron corpus) are shown in Table 4.5. As can be seen, these features are very specific to the Enron domain.

Feature mapping

The feature mapping approach attempts to find similar features in source and target domains and transform the features from the source domain to their similar feature in the target domain before training the commitment classifier. Features are considered to be word n-grams ($1 \leqslant n \leqslant 3$). The main assumption of the feature mapping method is that for each feature in the source domain, there is a feature with a similar meaning in the target domain. Therefore, for each domain specific feature in S, there is a domain specific similar feature in T and our goal is to find these features with similar meanings. For example, the similar feature of Enron in the Enron corpus would be Avocado in the Avocado corpus, as these two features are the names of companies these corpora belong to.

We assume that we have a set of emails in each domain. We extract sentences from the emails and, for each domain, we learn a semantic space in which each feature is represented as a low dimensional neural embedded vector. We use Word2Vec (the Skipgram architecture) [138] to generate the embeddings of features. Finally, we learn a linear transformation between the embedding spaces of source and target domains and use it for transforming features between domains.

Linear mapping-based transformations of embeddings between spaces were previously used for translation [137] and detecting semantic shifts [8]. In this approach, we first pick a set of words as anchors between domains as training samples for learning the mapping. We use a set $STOP$ of stopwords occurring in both source and target domains as anchors because they should have the same meaning in both domains and they can serve as fixed points around which features with varying meanings (usages) are located. We use a standard stopword list with a few additional words added (very frequent words in the corpus). Using the training samples, the goal is to learn a transformation matrix W^{ST} from domain S to domain T that minimizes the distance between the words and their mapped vectors.

The objective function to minimize is:

$$\underset{W^{ST}}{\operatorname{argmin}} \sum_{w \in STOP} \left\| W^{ST} V_w^S - V_w^T \right\|^2 . \tag{4.1}$$

V_w^S and V_w^T are the embeddings of w in the embedding spaces created for source and target domains, respectively. We use the gradient descent algorithm [168] to optimize the objective function. Using the learned transformation, the mapping of a feature w_S from source domain in target domain is determined as follows:

$$M(w) = \underset{w_T \in F_T}{\operatorname{argmax}}\, cos(W^{ST} V_w^S, V_{w_T}^T),$$

where cos is the cosine similarity and F_T is the set of all features (n-grams) in domain T.

4.4.2 Sample-level adaptation

As a sample-level adaptation method, we use the importance sampling approach. Importance sampling is a technique in statistics to estimate the parameters of a distribution (target distribution) given samples generated from a different distribution (source distribution) [172]. This technique has been applied to domain adaptation for classification [150]. Assume there are two distributions: $P_S(x, y)$ from which samples in the source domain are generated, and $P_T(x, y)$ from which samples in the target domain are generated. In our setting for the commitment detection task, x is a sentence and y is its corresponding commitment label. The goal is to create a model using labeled samples from S while optimizing the objective (e.g., the classifier loss) for samples in T. The importance sampling approach for classification works as follows. As mentioned, given a set of samples X and their corresponding labels Y, the goal is to find parameters of the classifier that minimize the loss for the samples in T:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathbb{E}_{P_T} l(X, Y, \theta),$$

where θ is parameters of the classifier and $l(X, Y, \theta)$ is the loss of the classification. We can rewrite the above equation as follows:

$$\begin{aligned}
\theta^* &\approx \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} P_T(x_i, y_i) l(x_i, y_i, \theta) \\
&= \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{P_T(x_i, y_i)}{P_S(x_i, y_i)} P_S(x_i, y_i) l(x_i, y_i, \theta) \\
&= \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{P_T(x_i) P_T(y_i|x_i)}{P_S(x_i) P_S(y_i|x_i)} P_S(x_i, y_i) l(x_i, y_i, \theta) \\
&\approx \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{P_T(x_i)}{P_S(x_i)} P_S(x_i, y_i) l(x_i, y_i, \theta)
\end{aligned}$$

Note that the assumption in the derivation from the third to the last step is that $P_S(y_i|x_i) = P_T(y_i|x_i)$, which implies that the conditional probability of classes given the samples is independent of the domain. $\frac{P_T(x_i)}{P_S(x_i)}$ is often referred to as the importance weight. Given the above derivation, we can still train a classifier using the samples from the source domain, however we need to weight the loss on samples by their importance weight. $P_S(x_i)$ and $P_T(x_i)$ are the marginal distributions of samples in the source and target domains respectively. To find θ^*, we need samples in the source side and also an estimation of $P_S(x_i)$ and $P_T(x_i)$.

To estimate $P_S(x_i)$ and $P_T(x_i)$, we use sets of samples from the source and target domains. We consider the target side as positive class and source side as negative class. Then, we train a domain classifier to predict for each sample how likely it is to be generated from target domain. We use a simple LR model to train the classifier. For each sample x_i, $P_T(x_i) = p$ and $P_S(x_i) = 1 - p$, where p is the probability that x_i is generated from T assigned by the trained domain classifier to x_i.

4.4.3 Deep autoencoder

Deep autoencoders have been successful in unsupervised feature extraction and representation learning [19]. Since these models are unsupervised, they attempt to model the underlying distribution from which the data is generated. In a transfer learning setting, autoencoders are used to learn a representation for a combined set of samples from source and target domains, thereby aiming to simultaneously represent samples from both domains, and yielding a model that should be domain independent [36]. After learning the representations model, encodings of samples in the source domain, with their labels, are used for training a classifier.

We first formulate the problem and then describe our proposed model by introducing its general architecture. We then explain the details of various parts of the architecture including input module, sample representation, output module, and loss functions.

The setting is as follows. We are given a set of commitment sentences from the source domain $X_S = \{(x_i^S, y_i^S)\}_{i=1}^N$ and a set of commitment sentences from the target domain $X_T = \{(x_i^T, y_i^T)\}_{i=1}^M$. We train the autoencoder embeddings using unlabeled examples from the source and target domain, and subsequently we apply a set of labeled target examples in addition to the source examples for training the sample representations and output module. Our goal is to use both X_S and X_T to create a domain independent classifier that has good performance in both domains. Our proposed model takes a candidate sentence from an email and predicts how likely that sentence constitutes a commitment between the email sender and the recipient. To do this, we introduce three different loss functions: reconstruction loss L_R, commitment loss L_C, and domain loss L_D. The reconstruction loss corresponds to how well the learned representations represent the samples. The commitment loss is the main objective which is included to minimize the errors of the commitment classifier. Finally, the domain loss is included to remove the domain bias from samples. In the training process our goal is to maximize the loss of the domain classifier, to avoid capturing domain-specific information during learning the sentence representation. Given the described loss functions, the final objective function of the proposed model is as follows:

$$L = \alpha L_R + \beta L_C - \gamma L_D, \tag{4.2}$$

where α, β,and γ control the effect of each loss function on the final loss and they are set based on some preliminary experiments explained in Section 4.6.4. The details of each loss function are explained in this section.

The overall structure of the proposed model is shown in Figure 4.2. The model contains three primary components:

Input module that provides a set of functions for encoding each input sentence x_i to a sequence of dense vectors $\{z_i^j\}_{j=1}^n$ where $z_i^j \in \mathbb{R}^d$ corresponds to embedding of the j-th word in the sentence and n is the number of words in the sentence; for simplicity, we assume that d is the dimension of the representation vector of words.

Sample representation that given the outputs of the input module (the sequence of embeddings of words), and learns a representation for the input sentence. The output of this module is a vector: $s \in \mathbb{R}^{d1}$, which can be considered as the aggregated representation of the input sentence. $d1$ is the dimension of the representation vector for samples. We will describe the details of this module in this section.

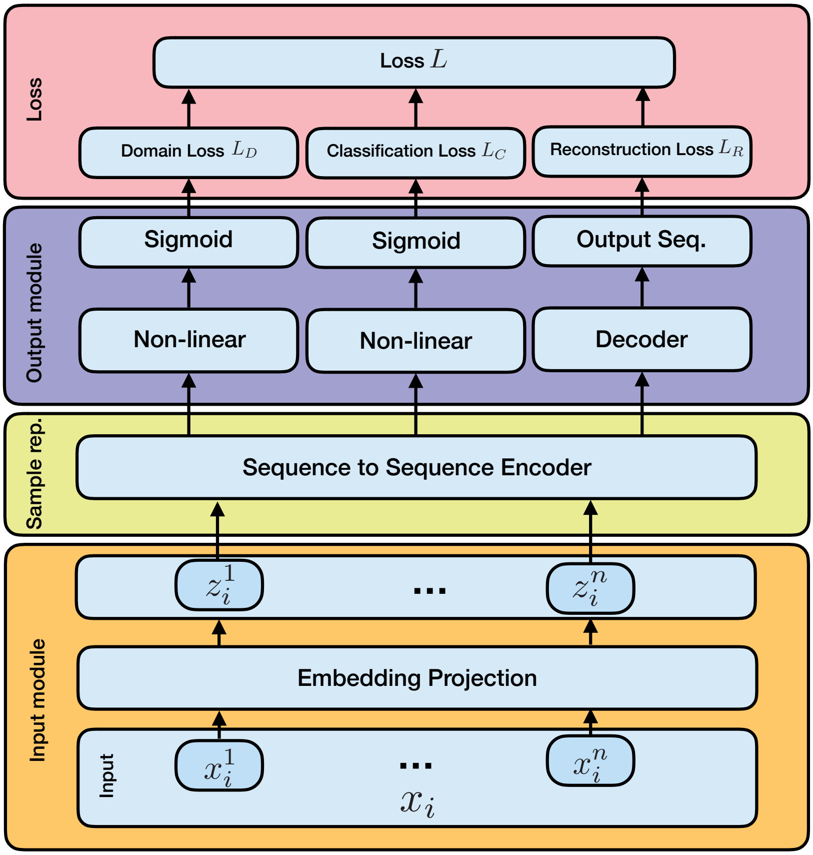


Figure 4.2: General schema of the proposed neural autoencoder model used for commitment detection.

Output module that captures how likely the input sentence constitutes a commitment based on the representation of the sentence that is provided by the previous module. The details of this unit are explained in this section.

In the following, we explain the input representation, output modules, and loss functions and how these modules are connected.

Input module

Input module projects an input sentence to a sequence of dense vectors with dimension d using a trainable embedding layer. Each vector in this sequence corresponds to a word in the sentence.

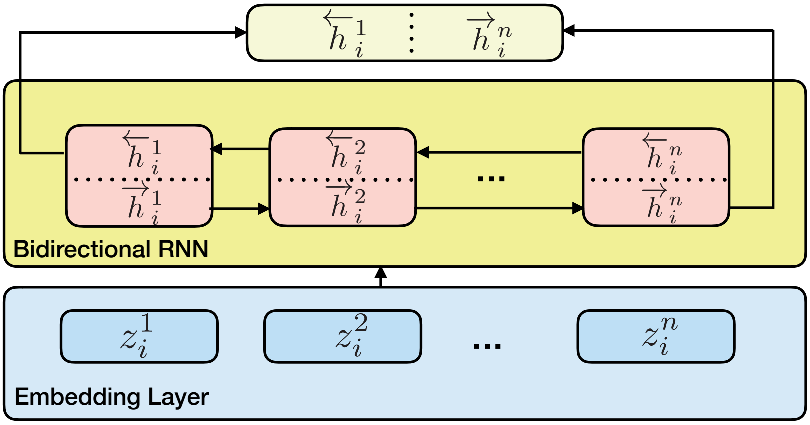


Figure 4.3: Architecture of the sequence to sequence encoder function in the input representation module.

Sample representation

To represent samples, we use a sequence to sequence recurrent neural network (RNN) as the encoder in our model. The RNN reads the input sequence $Z_i = [z_i^1, z_i^2, \ldots, z_i^n]$ in the left-to-right direction in the forward pass. It creates a sequence of hidden states, $[\overrightarrow{h}_i^1, \overrightarrow{h}_i^2, \ldots, \overrightarrow{h}_i^n]$, where $\overrightarrow{h}_i^j = \text{RNN}(z_i^j, \overrightarrow{h}_i^{j-1})$ is a dynamic function for which we can use, for example, an LSTM [83] or a GRU [38]. In this study, we use an LSTM for learning the representation. The RNN backward pass reads Z_i in the reverse direction, i.e., $\overleftarrow{h}_i^j = \text{RNN}(z_i^j, \overleftarrow{h}_i^{j+1})$, resulting in a sequence of hidden states $[\overleftarrow{h}_i^n, \overleftarrow{h}_i^{k-1}, \ldots, \overleftarrow{h}_i^1]$. We take the concatenation of the last hidden state of the forward pass and the first hidden state of the backward pass of the RNN, i.e., $\varphi(x_i) = [\overrightarrow{h}_i^k; \overleftarrow{h}_i^1]$, as the final representation for the given data field (see Figure 4.3).

Output module

The proposed architecture has three output modules: decoder output, commitment label, and domain label. In this section, we describe each output module.

Decoder output and reconstruction loss. The decoder is an RNN which, given the learned representation for the input sequence (output of the encoder), attempts to generate the input sequence. The goal of the decoder is to estimate the probability $P(o_i|x_i) = P(o_i^1, ..., o_i^{n'}|x_i^1, ..., x_i^n)$, where o_i^t is the output of the decoder at time step t. The decoding starts by reading a special symbol (‘GO’) at the first time step. Decoding stops by reading another special symbol (‘EOS’) at the end of each input sentence. Given the outputs at each time step we can determine the decoder’s output as follows: $P(o_i|x_i) = \prod_{t=1}^{n'} P(o_i^t|\varphi(x_i), o_i^1, ..., o_i^{t-1})$. We train the decoder, in an end-to-end training process in which, given mini-batches of samples $B = \langle x_i, y_i, d_i \rangle$, where y_i and d_i are commitment label and domain label of a sample x_i, we maximize the conditional log-likelihood of a correct output o_i given the input sequence x_i:

$$L_R(x_i) = -\sum_{i=1}^{|B|} \log(o_i|x_i) \tag{4.3}$$

Commitment label and classification loss. The commitment classifier is a feed-forward layer with tanh non-linearity, followed by a sigmoid. It receives the learned representation for the input sentence and predicts the probability of it constituting a commitment:

$$\begin{aligned} O_C &= \tanh(\boldsymbol{W}^{(C)}\varphi(x_i) + \boldsymbol{b}^{(C)}) && \in \mathbb{R}^{d_C} \\ \hat{y}_i &= \text{sigmoid}(\boldsymbol{w}^T O_C) && \in \mathbb{R}, \end{aligned}$$

where $\boldsymbol{W}^{(C)} \in \mathbb{R}^{d_C \times d1}$ and $\boldsymbol{b}^{(C)} \in \mathbb{R}^{d_C}$ are a trainable projection matrix and bias respectively, and d_C is the size of projection, and $\boldsymbol{w} \in \mathbb{R}^{d_C}$ is a trainable vector. The commitment classifier is trained in an end-to-end training process. Given mini-batches of data $\langle x_i, y_i, d_i \rangle$, we first predict $\hat{y}$ and calculate the loss using the cross-entropy loss:

$$L_C = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i) \tag{4.4}$$

Domain label and domain classification loss. As with the commitment classifier, we use a feed-forward layer with tanh non-linearity, followed by a sigmoid, as the output of the domain classifier. The domain classifier predicts the probability of the sentence being generated from the target domain:

$$\begin{aligned} O_D &= \tanh(\boldsymbol{W}^{(D)}\varphi(x_i) + \boldsymbol{b}^{(D)}) && \in \mathbb{R}^{d_D} \\ \hat{y}_i &= \text{sigmoid}(\boldsymbol{w}^T O_D) && \in \mathbb{R}, \end{aligned}$$

where $\boldsymbol{W}^{(D)} \in \mathbb{R}^{d_D \times d1}$ and $\boldsymbol{b}^{(D)} \in \mathbb{R}^{d_D}$ are a trainable projection matrix and bias, and d_D is the size of projection, and $\boldsymbol{w} \in \mathbb{R}^{d_D}$ is a trainable vector. The domain classifier is also trained in an end-to-end training process. We again first predict $\hat{d}$ and calculate the loss using the cross-entropy loss:

$$L_D = -d_i \log \hat{d}_i - (1 - d_i) \log(1 - \hat{d}_i). \tag{4.5}$$

4.5 Experimental Setup

We aim to understand how domain differences can affect the performance of email commitment detection models and how we can use a transfer learning approach to overcome any performance degradation due to these differences. Our goal is to answer **RQ3**. To this end, we break down **RQ3** to four research questions:

RQ3.1 Can commitments be reliably detected in emails?

RQ3.2 How does the performance of commitment models change when they are tested on a different domain than they are trained on? Can we reliably train a model on one domain and use it to detect commitments on a different domain?

RQ3.3 How can we characterize differences between domains and use this characterization for transferring knowledge between domains?

RQ3.4 Does the proposed autoencoder help to detect commitments more accurately?

RQ3.1 is concerned with the quality of automatic commitment detection models. To answer this research question, we use the collected datasets and train and test commitment detection models on the same domain and analyze their performance.

RQ3.2 is concerned with the effect of domain difference on the performance of commitment detection models. To answer **RQ3.2**, we train and test commitment models across domains and analyze their performance.

To answer **RQ3.3**, we try to characterize differences between domains and use this characterization to remove domain-specific bias from commitment models. We evaluate how successful these approaches are in characterizing the differences.

RQ3.4 is concerned with the performance of the proposed autoencoder model in commitment detection and its ability to learn domain independent representations for samples. We report the results of the autoencoder model and compare them to the results of the feature-level and sample-level adaptation approaches.

4.5.1 Evaluation metrics

We use standard evaluation metrics for classification such as the area under the receiver operator characteristic curve (AUC), precision, recall, and F1 measure.

4.5.2 General setting and hyperparameters

We set the number of hidden layers of the LSTM (forward and backward) model, d, $d1$, d_C, and d_D to 128. We set the initial learning rate to 10^{-3}. The batch size is set to 128. Training consists of 250K steps. Dropout of the LSTM model is set to 0.2. In training all commitment models, we perform five-fold cross validation and report the average of performance on five folds as the performance of the models.

4.5.3 Baselines

Since commitment detection is a relatively under-studied task, there are not many baselines to compare our proposed approaches to. Previous work in this area [41, 50, 102, 114, 115] either did not propose a model to automatically detect commitments or used very simple word/POS n-grams based features to train a model. We also use word and POS n-grams to create LR models for this task. In that sense, the LR model is our main baseline. However, our main goal is to show that the domain difference can be very problematic in creating commitment models and previously proposed n-gram based approaches fail to remove domain bias.

4.5.4 Statistical significance

For statistical significance testing, we compare our methods to baselines using paired two-tailed t-tests. We set α (the desired significance level) to 0.05. In Section 4.6, $^\blacktriangle$ and

▾ indicate that the corresponding method performs significantly better or worse than the corresponding baseline, respectively.

4.6 Results

Following the four research questions described in Section 4.5, we report the results of our proposed commitment detection methods.

4.6.1 Commitment detection results

To answer **RQ3.1**, we use the datasets described in Section 4.3.2 and train and evaluate commitment classifiers using LR. In this set of experiments, we only focus on the performance of the models trained and tested on the same domain. Our goal is to evaluate whether or not commitments can be detected automatically in emails, and whether the LR model can capture the commitment language in emails. Table 4.3 shows the performance of LR models trained for detecting commitments in Avocado and Enron datasets using word n-gram and POS n-gram representations. The commitment models achieve a reasonable performance. This result indicates that commitments can be reliably detected in emails. There was no significant difference between the performance of models trained on word n-gram and models trained on POS n-grams, and in the remainder of this chapter we only report the results based on the word n-gram representation as this representation is more efficient and has lower dimensionality.

Table 4.3: Results of LR method for detecting commitments trained and tested on the same domain.

Dataset	**Precision**	**Recall**	**F1**	**AUC**
Avocado (word n-grams)	0.82	0.81	0.81	0.86
Avocado (POS n-grams)	0.82	0.84	0.82	0.86
Enron (word n-grams)	0.80	0.77	0.78	0.88
Enron (POS n-grams)	0.79	0.78	0.78	0.87

4.6.2 Cross-domain results

To answer **RQ3.2**, we evaluate the performance of commitment models on a different domain than they are trained on. Again, we use LR for training commitment models. Table 4.4 shows the performance of trained models across domains. The results show that the performance of commitment models degrades when moving across domains in terms of almost all used evaluation measures, and we cannot reliably train a commitment model on one domain and use it to detect commitments on a different domain. For the Avocado→Enron case the precision drops more than recall (precision drops from 0.82 to 0.74 and recall drops from 0.81 to 0.78). For Enron→Avocado case, the opposite is true. The primary reason for higher drops in precision in the Avocado→Enron case is that the Avocado dataset contains more positive samples, so the trained model is more inclined towards assigning positive labels to the samples. However, the Enron dataset contains more negative samples. So, for the Avocado→Enron case, the false positive rate is high, which leads to lower precision. For Enron→Avocado the false negative rate is high (as the Enron dataset has more negative samples and the trained classifier is more inclined towards assigning negative labels to samples), which leads to lower recall.

Table 4.4: Performance of LR method across domains.

Train	**Test**	**Precision**	**Recall**	**F1**	**AUC**
Avocado	Avocado	0.82	0.81	0.81	0.86
	Enron	$0.77^{\blacktriangledown}$	$0.69^{\blacktriangledown}$	$0.73^{\blacktriangledown}$	$0.67^{\blacktriangledown}$
Enron	Enron	0.80	0.77	0.78	0.88
	Avocado	$0.74^{\blacktriangledown}$	0.78	$0.76^{\blacktriangledown}$	$0.58^{\blacktriangledown}$

4.6.3 Characterizing inter-domain differences

Next, we answer **RQ3.3** by training models that can detect and characterize the differences between email domains and use this characterization for removing domain bias from commitment models before training them. We first design a binary classifier which attempts to identify the source domains for samples. The input of the classifier is a sentence and the output is the domain label for the sentence. We take 5,000 samples from each of Avocado and Enron datasets and represent them using a bag-of-words n-gram representation. Then, we train an LR model to predict domain labels for samples. Figure 4.4 shows the precision-recall curve of the domain classifier. The classifier achieves an F1 score of 0.85. Table 4.5 illustrates a sample of features that are strongly associated with the Enron domain. This result indicates that there is a domain bias in the samples and even a simple LR classifier can characterize differences between domains.

Table 4.5: The most informative unigram features indicating the Enron domain.

"enron", "gas", "ena", "houston", "ferc", "eol", "energy", "ees", "counterparty"

We use this characterization to remove domain bias from source and target datasets before training commitment models. Table 4.6 shows the performance of models that use the characterization of the difference between domains for training domain-independent models. Three observations can be made from the results. First, all transfer learning models improve the performance of the baseline LR model, indicating that we can remove the domain bias using the transfer learning approach. The improvements of all methods are statistically significant over the LR method for Avocado→Enron. However, on Enron→Avocado only the importance sampling method achieves significant improvements over the LR method in terms of all metrics. This result indicates that the transfer learning approach is more successful in removing the domain bias from the Avocado dataset. Second, the linear mapping approach has a slightly better performance than the importance sampling method for Avocado→Enron. Based on our

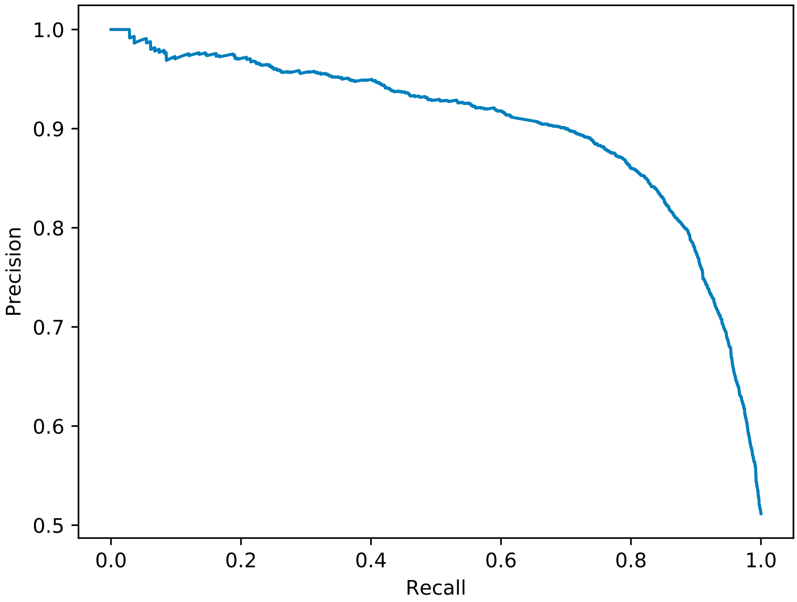


Figure 4.4: The precision-recall curve of the domain classifier (predicting source domain of the samples) (F1 score = 0.85).

analysis the quality of the mapping created from Avocado to Enron is higher than the quality of the created mapping from Enron to Avocado. The average cosine similarity of the mapped embedding of words from Avocado embedding (to Enron embedding) to their embedding in Enron space is 0.78. This value for Enron→Avocado mapping is 0.69. The better quality of the mapping leads to better transformation from Avocado to Enron and better performance for Avocado→Enron. When mapping from Avocado to Enron, the words project to a more meaningful place in the Enron embedding space. Third, the importance sampling method achieves significant improvements for both Avocado→Enron and Enron→Avocado cases. This indicates that adaptation at the sample level is more effective for the commitment detection task. We use importance sampling as a baseline in Section 4.6.4 and compare its performance to the performance of the proposed deep autoencoder.

4.6.4 Transfer learning results

To answer **RQ3.4**, we evaluate the performance of the proposed autoencoder model in the commitment detection task and compare its performance to that of the importance sampling method. As an additional baseline, we use an LR model trained on a combination of samples in both the Avocado and Enron datasets. Table 4.7 shows the performance of the autoencoder model with different loss functions. The training of the autoencoder that uses all loss functions is done on the combination of both Avocado and Enron samples with their labels. We again perform five-fold cross validation on the test set. At each step, we use four folds in target side in addition to all samples on

Table 4.6: Performance of different domain adaptation methods for detecting commitments. IS: Importance Sampling, LM: Linear Mapping, FS: Feature Selection.

Train	Test	Method	Precision	Recall	F1	AUC
Avocado	Enron	LR	0.77	0.69	0.73	0.67
		IS	0.81▲	0.75▲	0.77▲	0.74▲
		LM	0.83▲	0.76▲	0.79▲	0.75▲
		FS	0.80▲	0.73▲	0.76▲	0.73▲
Enron	Avocado	LR	0.74	0.78	0.76	0.58
		IS	0.78▲	0.85▲	0.81▲	0.71▲
		LM	0.75	0.81▲	0.77	0.64▲
		FS	0.74	0.80▲	0.76	0.62

source side for training the model and evaluate the trained model on the fifth fold. For training AE_R and AE_{R+D} we train the models using unlabeled samples in the source and target sides. Then, we use the trained model to represent labeled samples from the source side. Finally, we train an LR model using these representations and evaluate its performance on the target side.

The results show that each introduced loss function contributes to the performance of the autoencoder, and using all loss functions achieves the best performance. Based on some preliminary experiments we set $\alpha = 0.1$, $\beta = 0.6$, and $\gamma = 0.1$ in Equation 4.2. This indicates that the commitment loss has more effect on the performance of the model, while other losses have the same contribution. The proposed model outperforms both IS and the LR model trained on a combination of the samples from source and target side. This result shows the ability of the proposed autoencoder model to remove the domain bias from data and achieve a robust model.

To observe the effect of the size of training set on the performance of the proposed autoencoder model, we design an experiment in which we vary the number of samples in the target side and measure the performance of the model. Figure 4.5 shows the results of this experiment. The results show that adding more samples in the target side boosts the performance of the model on both datasets. With about 50% of the samples, the model already achieves a good performance. For the Enron case, after having seen 50% of the data, adding more samples does not affect the performance of the model significantly. However, for Avocado this is not the case. The main reason is that Enron is a significantly larger dataset and with only 50% of the data the model can already generalize, while Avocado is smaller and adding more samples helps learn a better model.

4.7 Discussion and Implications

We have shown that there is a significant degradation in the performance of commitment detection models across domains (email corpora), that the differences can be accurately characterized, and that domain adaptation, specifically transfer learning, can help ameliorate domain biases and yield performance improvements.

We evaluated the performance of the methods on two publicly available email

Table 4.7: Performance of the proposed autoencoder for domain adaptation for detecting commitments. IS: Importance Sampling, AE_R: Autoencoder with only reconstruction loss, AE_{R+D}: Autoencoder with reconstruction and domain losses, AE_{All}: Autoencoder with reconstruction, domain, and classification losses. The baseline to perform significance tests is IS.

Train	**Test**	**Method**	**Precision**	**Recall**	**F1**	**AUC**
Avocado	Enron	IS	0.81	0.75	0.77	0.74
		LR	0.80	0.79	0.79	0.78
		AE_R	0.80	0.78$^{\blacktriangle}$	0.79	0.76
		AE_{R+D}	0.81	0.79$^{\blacktriangle}$	0.80$^{\blacktriangle}$	0.77$^{\blacktriangle}$
		AE_{All}	0.82	0.81$^{\blacktriangle}$	0.81$^{\blacktriangle}$	0.79$^{\blacktriangle}$
Enron	Avocado	IS	0.78	0.85	0.81	0.71
		LR	0.80	0.82	0.81	0.70
		AE_R	0.77	0.82	0.79	0.68
		AE_{R+D}	0.77	0.84	0.80	0.69
		AE_{All}	0.79$^{\blacktriangle}$	0.87$^{\blacktriangle}$	0.83$^{\blacktriangle}$	0.72

datasets: Avocado and Enron. The use of public data sets improves the replicability of our study and is sufficient to demonstrate the value of domain adaptation for commitment detection.

The created commitment dataset based on the Enron collection has far fewer positive samples. To have more positive samples, we biased the Avocado dataset. Generating the Avocado dataset in the same way as the Enron dataset would allow us to better understand the models and their transferability, however, creating such a large dataset is costly and beyond the scope of this study.

The designed methods differ a lot in their efficiency. The deep autoencoder model required about 12 hours to train on average and about 50 milliseconds to extract a representation for a test sample, while on the same machine the importance sampling methods takes about 4 minutes to train and several microseconds to test on average. Although in training phase the autoencoder is quite slow, it is more efficient during testing.

We attempted to explore the effectiveness of models that work in different granularities and can be applied easily to the task. The demonstrated viability of transfer learning is promising. The effectiveness of other domain adaptation methods such as [45], and alternatives such as multi-task learning models [130], needs to be explored. Beyond commitments, the value of transfer learning for other detection tasks in email, e.g., extracting requests and detecting task completion [68, 102], should be investigated, as should the applicability of these methods to media beyond email (e.g., SMS, instant messaging, meeting transcripts). In this chapter, we applied our models to transfer knowledge between two domains. However, the proposed approaches are easily extensible to consider more than two domains.

The transfer learning methods used in this study (except for importance sampling, feature selection, and linear mapping), require labeled samples in the target domain as well as the source domain. In applying these methods, additional data collection efforts would be required to obtain these additional labels and the cost of that should be factored into decisions regarding their application. For reference, the autoencoder

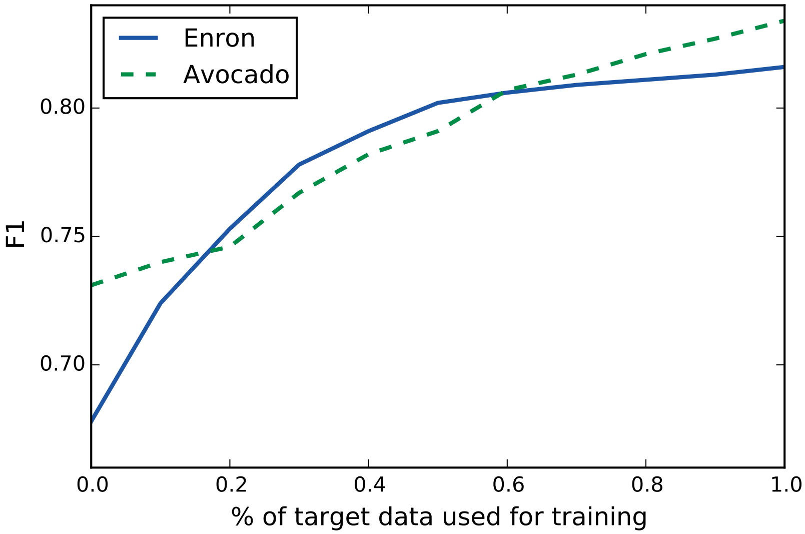


Figure 4.5: The effect of size of training samples in target domain on the performance of the proposed autoencoder.

model, which uses all loss functions, achieves a good performance with just a few thousand labeled samples in the target domain.

Future work involves using the domain adaptation methods in practice to train more domain independent models and deploying them as skills and add-ins inside digital assistants and task management tools, respectively. Deployment would enable the computation of online performance measures based on implicit and explicit feedback from users making the commitments (rather than offline labeling from third-party judges).

Having presented our approaches to manage documents automatically from different angles, we now focus on the second part of the thesis and show how metadata and structure can be helpful in exploratory search.

Part II

Using Structure and Metadata for Exploratory Search

5
Computing Semantic Shifts in Political and Media Discourse

In the second part of the thesis, we focus on using structure and metadata in exploratory search tasks. As the first task, we focus on measuring semantic shifts and study how metadata can be helpful in measuring semantic shifts in the meaning of words. In this chapter, our main aim is to answer **RQ4**.

5.1 Introduction

Words are always ‘under construction,’ their meaning is unstable and malleable [118, 175, 187, 210]. Semantic fluctuations can result from a concept’s ‘essentially contested’ nature. “What does democracy mean?” or “what values are democratic?”. The answer changes according to the ideological perspective or *viewpoint* [67] of the person uttering the term. Equally important is the influence of historic events. The understanding of ‘terrorism,’ for example, has significantly changed as a result of the 9/11 attacks [25, 164]. Currently, only a few studies have attempted to compute the ‘malleability of meaning’ and monitor semantic shifts [75, 90, 104, 110]. Most (if not all) of these approaches have focused their efforts on uncovering change over time. However, there are other valuable dimensions that can cause semantic shifts such as social or political variability.

As an example, Figure 5.1 shows the semantic shifts over two dimensions: time and political context, i.e., membership of a parliamentary party at the British House of Commons. The speeches given by the members of each party are used for constructing their corresponding semantic spaces. This can be extended to social parties or groups of like-minded people in social forums such as Facebook. The first example in the figure (the word “moral”) shows that a semantic shift can occur over time and across different contexts. However, as the second example shows, although the meaning of a word (such as “democracy”) can stay stable over time, it can still differ between certain groups. Therefore, social context is another valuable dimension that can explain semantic shifts. In this study, we explore the *semantic stability* of words by computing how contextual factors, such as social background and time, shapes—or, at least reflects—shifts in

This chapter was published as [8].

meaning.

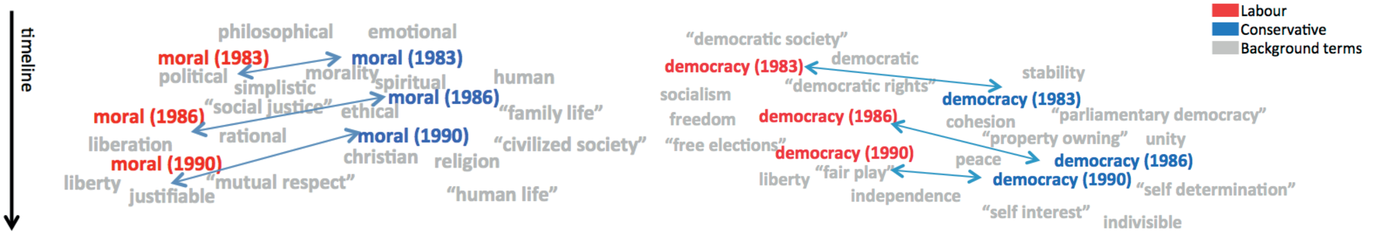
We first use distributional semantics to generate embedding spaces from categorized corpora, where a category can be a certain context (such as speeches given by a political party). In the example given in Figure 5.1, there are two categories: Conservative and Labour parties. Then, we propose different approaches to compare the vector representations of words between spaces. In the remainder of this chapter, we define each of these categories as *viewpoints*, since they reflect the semantic constellation of terms from a specific social perspective. In this study, we only consider two viewpoints. However, our approaches are easily extendable to multiple (i.e., non-binary) viewpoints. The challenging part of this task, and the main contribution of this study, is to develop techniques that compare vectors across spaces with different dimensionality structures. We consider three methods for comparing meaning across vector spaces: (1) Inspired by [137], we create a linear mapping between two embedding spaces, project words from one embedding space to the other and measure whether the projected word lands closely to the word in the other space. (2) Inspired by [104], for each viewpoint, we construct a graph such that the nodes are words and edges are the similarities between them. Then, using graph-based similarity measures we compute how similar the neighbors of a word in two embedding spaces are. And (3) We define a measure that combines these two measures.

As stated, in this work, our *main research problem* is to study how semantic shifts in words are happening not just over time but also across social dimensions, quantify the size of shifts, and explore applications that can benefit from information about shifts. We evaluate the proposed approaches in three different tasks: measuring semantic shifts, document classification, and contrastive viewpoint summarization. Our aim is to answer **RQ4**:

> *How can we use metadata information to measure semantic shifts? How effective are the proposed approaches in quantifying the changes in word meaning over various dimensions such as time and political context?*

Our main contributions are: (1) We show that semantic shifts not only occur over time, but also across different viewpoints in a short period of time. (2) We improve the linear mapping approach [137] for detecting semantic shifts and propose a graph-based method to measure the size of semantic shifts in the meanings of words. (3) We employ word stability measures in contrastive viewpoint summarization and document classification and extensively evaluate our proposed approach to these tasks. (4) Our analysis shows that the two laws of semantic change proposed in [75] hold for semantic shifts across viewpoints. Moreover, we introduce a new law of semantic change which implies that concrete words are less likely to shift meaning while abstract words are more likely to do so. (5) We make the evaluation dataset for detecting semantic shifts and contrastive viewpoint summarization publicly available.[1]

[1] The datasets are available here: `http://dx.doi.org/10.7910/DVN/BJN7ZI`.



(a) Shift over both time and political context dimensions

(b) Shift over ONLY political context dimension

Figure 5.1: Visualization of semantic shifts in meaning of words "democracy" and "moral" over time and along Conservative and Labour parties in the UK parliament. The approach proposed in [75] is used for visualization. (a) The meaning given by Labours to "moral" is shifted from a "philosophical" concept to a "liberal" concept over time. In the same time, the meaning of this word is shifted from a "spiritual" concept to a "religious" concept from Conservatives' viewpoint. Moreover, two parties gave a very different meanings to this word. (b) The meaning of "democracy" is stable over time for both parties. However, Conservatives refer to democracy mostly as a "unity" concept, while Labour associate it with "freedom" and "social justice."

5.2 Related Work

In this section, we review related studies from three perspectives: detecting semantic shifts, methods for detecting ideology and approaches to viewpoint summarization.

5.2.1 Detecting semantic shifts

With the appearance of Word2Vec [136] and GloVe [154], unsupervised methods have become increasingly popular as tools for generating a vector representation of words. Notwithstanding the popularity of these vector representations, relatively few studies have attempted to compare embeddings generated from different corpora. The approaches closest to our approach are [75, 90]. Jatowt and Duh [90] create time-stamped word representations per decade, and use these to monitor semantic fluctuations over more than 400 years. Words are represented as high dimensional vectors in which the values indicate how often a word co-occurs in the close vicinity of the target word. Hamilton et al. [75] use orthogonal Procrustes to align embeddings learned for different time-periods. They show that using a linear transformation is effective to find semantic shifts over time. Moreover, based on their proposed method for measuring semantic shifts over time, they propose two laws of semantic change, namely the law of Conformity and the law of Innovation. Similar to these works, other studies also tried to capture semantic shifts in the meaning of words over time [76, 82, 104, 110, 220] and also in the meaning of loanwords [183].

An alternative to our embedding based approach would be to use a direct high-dimensional representation of the co-occurring terms as in e.g., [90, 123], which retains the dimensionality structure and allows a direct comparison across vector spaces. However, given that we want to detect semantic differences, it would be unrealistic to assume that all the dimensions mean the same in both corpora.

Another relevant line of study is monitoring and tracking events and topics over time [87, 126, 190]. These approaches are aimed at detecting a set of topics and monitoring their change over time. Our approach is different from topic tracking methods as we do not restrict ourself to monitoring a limited set of topics. For evaluating the proposed approach, similar to previous work in detecting semantic shifts over time [75, 76, 90, 104, 110], we select a small set of words whose meaning shifted and evaluate how the proposed approach is successful in detecting them. In some other studies [16, 191], distributional semantic approaches are combined with structural data (WordNet and knowledge bases) to track lexical changes.

5.2.2 Ideology and political text classification

Besides monitoring changes in meaning, this study demonstrates how knowledge about semantic shifts contributes to other tasks such as the classification of political texts. Kusner et al. [111] applied word embeddings to calculate the distance between documents and utilized these estimated distances for classifying documents. Their results show that embedding-based approaches to document classification outperform others such as LDA and LSI. Similar to previous work, we utilize word vectors for text classification. Our task differs, however, since we employ multiple embeddings to enhance classification performance.

We use political text to evaluate the proposed approaches. Previous approaches to political text classification [51, 53, 55, 81] are largely limited to word counting—or other units such as syntactic rules—thereby ignoring the adversarial semantics that characterizes political discourse. Using word embeddings we attempt to capture this 'macrocosmos' of political ideas.

Our approach for document classification is to use word embeddings and expand documents using extracted associations that are specific for each class. This kind of document expansion helps in resolving the vocabulary mismatch issue and increasing the discrimination between different classes. Using word embeddings was shown to be very effective to boost the performance in document classification [97, 111, 214].

5.2.3 Cross-perspective opinion mining and summarization

Even though opinions can be extracted at the word, sentence, or document level [151], they are usually represented as topics. Most of the current approaches rely on topic models and jointly extract topic-opinion pairs from a set of unlabeled documents [127, 134, 167]. They consider, however, only one point of view about topics. To extract contrastive opinions about topics, previous research [64, 106, 186] proposed to jointly extract topics and opinions coming from different viewpoints. Besides extracting a polarity score of each viewpoint about topics, these approaches also summarize the opinions about topics. Similarly, we also perform *contrastive viewpoint summarization* and for each topic we estimate a score which expresses and summarizes differences in word meaning. However, instead of opinions, we estimate and summarize the diverging viewpoints on a concept. Viewpoints are different than opinions as they do not necessarily carry sentiment information.

5.3 Measuring Word Stability

In this section, we describe our approach for measuring semantic stability of words.

5.3.1 Task overview

We define semantic *word stability* as the similarity of (a word's) vector representation across viewpoints. A *viewpoint* is defined as a set of texts that share a specific metadata feature, for example texts generated by a social entity such as a political party. Words whose meaning is independent of perspective will obtain a high stability score—for example in a political context we expect conservatives and progressives to disagree on the concept of 'democracy', but not on the semantics of the word 'lettuce.' More formally, our method takes as input a word, and returns a number that expresses its meaning stability across viewpoints.

To measure the semantic stability of words, we first use distributional semantics to create a separate embedding space for each viewpoint. Then, using trained embeddings we map each word to a vector in their respective viewpoint. Finally, we compare the embeddings of each word in different embedding spaces. Our proposed approaches are applicable for words in the intersection of the vocabularies of two embeddings spaces. In the reminder of this section, we use V^0 and V^1 to represent the created embedding spaces for two viewpoints. V_w^i is the vector of word w in embedding space V^i.

Because the embedding spaces are different and have different dimensionality structures, we cannot compare the vectors of a word in two different spaces directly. In this study, we propose different approaches to address this issue. Below, we describe three methods for comparing words in different vector spaces: linear mapping, neighbor-based approach and, lastly a combination of the two.

5.3.2 Linear mapping

The application of linear transformation for translating vectors from one space to another was first proposed by [75, 137]. In this approach, a set of words with their extracted vectors in two embedding spaces are used to learn a mapping. Mikolov et al. [137] start with a set of training words (mainly function words, whose meaning should be stable irrespective of viewpoint or domain) in two embedding spaces. Using the training samples, the goal is to learn a transformation matrix W^{ij} from embedding space i to embedding space j that minimizes the distance between the words and their mapped versions. The transformation matrix is learned using the gradient descent algorithm. The objective function is:

$$\underset{W^{ij}}{\operatorname{argmin}} \sum_{w \in X} \left\| W^{ij} V_w^i - V_w^j \right\|^2, \tag{5.1}$$

where X is the set of training words. We denote the transformation matrix from embedding space V^i to embedding space V^j by W^{ij}. We use a standard stopword list with a few additional words added (very frequent words) to learn the transformation matrix. As the meaning of these words should in theory be similar in both time periods, they serve as fixed points in the mapping around which the words with varying meaning are situated. The transformations are learned on a total of 813 words from the stop list.

'Stability' measure of a word w, is then expressed by the following measure:

$$s_{lin}(w) = \frac{sim_{01}(w) + sim_{10}(w)}{2}, \tag{5.2}$$

where sim_{01} and sim_{10} are calculated using the following equation:

$$sim_{ij}(w) = cos(W^{ji} W^{ij} V_w^i, V_w^i), \tag{5.3}$$

where cos is the cosine similarity. The stability of a word using this measure equals to the similarity of its vector to its mapped vector after applying the mapping back and forth.

5.3.3 Neighbor-based approach

The second method for measuring word stability is based on the intuition behind graph-based node similarity measures. The similarity of two nodes in a graph is determined

```
Input: $V^0$: embedding space of viewpoint 0
Input: $V^1$: embedding space of viewpoint 1
Input: $T$: the number of iterations
Input: $\mathcal{V}$: the intersection of the vocabularies of $V^0$ and $V^1$
Result: $s^T_{nei}$: a vector containing the stability of words
for $w \in \mathcal{V}$ do
    $s^0_{nei}(w) = 1$
end
for $t \leftarrow 1$ to $T$ do
    for $w \in \mathcal{V}$ do
        $sim^t_{01}(w) = \frac{\sum_{w' \in N^1_w} cos(V^0_w, V^0_{w'}) s^{t-1}_{nei}(w')}{|N^1_w|}$
        $sim^t_{10}(w) = \frac{\sum_{w' \in N^0_w} cos(V^1_w, V^1_{w'}) s^{t-1}_{nei}(w')}{|N^0_w|}$
        $s^t_{nei}(w) = \frac{sim^t_{01}(w) + sim^t_{10}(w)}{2}$
        Min-Max normalize $s^t_{nei}$ to fall into [0,1] interval
    end
end
```

Algorithm 1: The algorithm for computing Neighbor-based stability of words. N^0_w is the set of most similar words to w in embedding space V^0 based on cosine similarity of words vectors.

by the similarity of their neighbors [91]. We consider each word in an embedding space as a node and its neighbors are the closest nodes to it, measured by cosine similarity. However, instead of one graph, we construct two graphs for two embeddings. For each word, we calculate the similarity of its neighbors in two different graphs and use this similarity as the stability of the word. This method assumes that words with similar meaning have similar neighbors. Thus, we can calculate stability by quantifying the extent to which words in different spaces still share neighbors.

Based on this assumption, we define an iterative method for calculating word stability. The algorithm is described in Algorithm 1. We first suppose that all words are stable and initialize $s^0_{nei}(w) = 1$ for all words. Secondly, depending on the depth parameter t, this method also takes into account the "neighbor's neighbor" etc. During the first iteration, only direct neighbors contribute to the stability of words. At iteration $t = k$, the indirect neighbors accessible by k edges in the graph contribute to the stability of words.

5.3.4 Combination: Co-occurrence of neighbors and linear mapping

The third, and last, stability-metric combines the neighbor-based approach with linear mapping. Each of these metrics provides different signals about the stability of a word: the linear mapping is solely based on the mapped vectors of the word while the neighbor-based approach is based on the vectors of neighbors and does not take into account the vector of the word itself. Thus, we combine these metrics to achieve better stability scores. This stability measure is based on the number of co-occurring neighbors *and* their similarity to the target word. The algorithm is described in Algorithm 2. For each word w, the weights of its neighbors reflect their place (or index) in a ranked list comprising the N most similar words to w. We define and combine two different stability signals: (1) $C^t_{ij}(w)$ represents the count of neighbors of word w in embedding V^i based on their index in the ranked list of neighbors of w in embedding V^j. $C^t_{ij}(w)$ is defined based on the words that are neighbors of word w in both embedding spaces. (2) $sim^t_{ij}(w)$ is based on similarity of mapped vectors from embedding space V^i to embedding space V^j and their vectors in space V^j, for the words that are neighbors of w in embeddings V^i but not in embedding V^j.

To give an example of how to compute $C^t_{ij}(w)$ in Algorithm 2, consider the following neighbor list:

$$N^0_w = [n1, n2, n3, n4, n5]$$
$$N^1_w = [n2, n4, n1, n5, n6]$$

Each neighbor in list N^0_w is obtained (if possible) from list N^1_w, along with the index. The final count after the first iteration ($C^0_{01}(w)$) then becomes: $C^0_{01}(w) = 5 * 4 - (2 + 0 + 1 + 3) = 14$. Note that this summation contains four terms instead of five, as neighbor $n3$ does not occur in list N^1_w. Therefore, in order to be able to take neighbor $n3$ into account when computing the agreement, the linear mapping is used to map the vector of $n3$ to a vector representing it in V^1. Then, the cosine similarity from the mapped vector to the target word vector is incorporated in calculating the stability value of w (using $sim^0_{01}(w)$). λ is defined as follows:

$$\lambda = \begin{cases} 1, & N^0_w = N^1_w \\ 0, & C^t_{01} = 0 \text{ and } C^t_{10} = 0 \\ 0.5, & \text{otherwise.} \end{cases} \tag{5.4}$$

5.4 Experimental Setup

We evaluate the performance of our approach intrinsically by detecting semantic shifts task (the details of this evaluation method is described in Section 5.4.3 and Section 5.5.1) and extrinsically in the document classification and viewpoint summarization tasks (the details of these evaluation methods are described in Section 5.4.4 and Section 5.5.2). Recall that **RQ4** is:

RQ4 How can we use metadata information to measure semantic shifts? How effective are the proposed approaches in quantifying the changes in word meaning over various dimensions such as time and political context?

We answer **RQ4** by breaking it down into four research questions:

RQ4.1 How effective are the proposed approaches in quantifying changes in word meaning over various dimensions such as time and political context?

RQ4.2 To what extent can these models improve performance on other tasks, such as document classification?

```
Input: $V^0$: embedding space of viewpoint 0
Input: $V^1$: embedding space of viewpoint 1
Input: $T$: the number of iterations
Input: $\mathcal{V}$: the intersection of the vocabularies of $V^0$ and $V^1$
Input: $\lambda$: the combination parameter determined by Equation 5.4
Result: $s^T_{com}$: a vector containing the stability of words
for $w \in \mathcal{V}$ do
    $s^0_{com}(w) = 1$
end
for $t \leftarrow 1$ to $T$ do
    for $w \in \mathcal{V}$ do
        for $i, j \leftarrow \{0, 1\} \wedge i \neq j$ do
            $C^t_{ij}(w) = |N^i_w| \times |N^i_w \cap N^j_w| - \sum_{w' \in N^i_w \cap N^j_w} \frac{rank_j(w')}{s^{t-1}_{com}(w')}$
            $sim^t_{ij}(w) = \frac{\sum_{w' \in \{N^i_w \setminus N^j_w\}} cos(W^{ij} V^i_{w'}, V^j_w) s^{t-1}_{com}(w')}{|N^i_w \setminus N^j_w|}$
        end
        $s_{nei}(w) = \frac{C^t_{01}(w) + C^t_{10}(w)}{2 \sum_{i=1}^{N_w} i}$
        $s_{lin}(w) = \frac{sim^t_{01}(w) + sim^t_{10}(w)}{2}$
        $s^t_{com}(w) = \lambda s_{nei}(w) + (1 - \lambda) s_{lin}(w)$
        Min-Max normalize $s^t_{com}$ to fall into [0,1] interval
    end
end
```

Algorithm 2: The algorithm for computing the stability of words based on combination of neighbor-based and linear mapping approaches. $|N_w|$ is the number of neighbors considered (i.e., 100), and $rank_j(w')$ is the rank that neighbor w'— which is an element of $N_i(w) \cap N_j(w)$— has in the ranked list of neighbors of w in embedding space V^j.

RQ4.3 How do the proposed approaches perform in summarizing different viewpoints expressed in two large corpora about different concepts?

RQ4.4 Do temporal laws of semantic change hold for shifts across viewpoints?

RQ4.1 is concerned with the quality of stability values estimated for words using different approaches. To answer **RQ4.1**, we construct an evaluation set and evaluate the accuracy of different approaches in measuring stability of words. In Section 5.5.1 the results of the experiments regarding **RQ4.1** are reported.

To answer **RQ4.2**, we use the stability values for document classification. We first expand the documents using the stability values and employ the expanded documents for classifying the speeches in the UK parliament to the parties. The details of this experiment are described in Section 5.4.4 and the results are reported in Section 5.5.2.

To answer **RQ4.3**, we utilize the word stability values for contrastive viewpoint summarization. We first generate the summary for a set of chosen words using different methods and ask human annotators to assess the summaries. The details of the evaluation process are described in Section 5.4.4. The results of experiments related to **RQ4.3** are reported in Section 5.5.3

RQ4.4 is concerned with the validity of laws of semantic shifts, namely the law of Conformity (implying that the rate of semantic change scales with an inverse power-law of word frequency) and the law of Innovation (reflecting that the semantic change rate of words is highly correlated with their polysemy) across viewpoint. To answer **RQ4.4**, we analyze the correlation of semantic shifts with their frequency, polysemy, and concreteness. The results of the experiments concerning **RQ4.4** are described in Section 5.5.4.

5.4.1 Datasets

To evaluate how effectively the methods described in Section 5.3 capture and summarize semantic shifts, we run multiple experiments using data sourced from the New York Times corpus[2] and the digitized proceedings of the British House of Commons— also referred to as the Hansard.[3]

Our corpus of political texts comprises the parliamentary and public speeches from the Thatcher years. This period contains 640,184 speeches. Within the broader context of British postwar politics, this era represents a break with the postwar Keynesian consensus, and was accompanied by a hardening division between left and right. In this work, we study how much the concepts in the Thatcher period have different meaning from a 'Conservative' and 'Labour' point of view.

The New York Times dataset contains 1,855,671 articles published between 1987 and 2007. We study how the meanings of words shifted after 9/11 in this newspaper. For example, as the terrorists involved in the 9/11 attacks were professors of Islam, it could be of value to investigate whether this had any affect on how Islamic faith is framed in media discourse. To do this, we divide the articles in the New York Times dataset into two viewpoints, i.e., articles before and after 9/11. We consider these two sets as two different viewpoints and study how the meaning of concepts are different based on these two point of views.

5.4.2 Preprocessing and general setting

We use Word2Vec [136] to generate word embeddings. We apply the Skipgram architecture and remove words with less than 20 occurrences. We train an embedding with 300 dimensions with a window size of 10.

Linear mapping refers to the linear transformation method introduced in Section 5.3.2. **Neighbor-based** method is the method introduced in Section 5.3.3 and **Combination** is the method described in Section 5.3.4. In estimating stability values using the Combination method, we set $|N_w^i| = 100$ which reflects that we only use top 100 closest words to each word for estimating the stability values. In Algorithm 1, for each word w, we again use the top 100 closest word to each word for estimating the stability values, however from this set we remove neighbors with similarity lower than 0.4 to w. For calculating stability values we set $T = 5$ (the number of iterations of the Neighbor-based method and the Combination method) since based on our experiments

[2] https://catalog.ldc.upenn.edu/LDC2008T19

[3] http://www.parliament.uk/business/publications/hansard/commons/

Table 5.1: The selected concepts for evaluating the word stability measures in detecting semantic shifts and summarizing viewpoints.

Detecting semantic shifts task (the UK parliament)	Summarization task (the New York Times)
privatisation, unemployment, working_class, society, homosexual, fairness, public_sector, justice, liberalism, communism, constitution, free_market, sovereignty, accountability, inequality, moral, conservatism, profit morality, tolerance, opponent, poor, bureaucracy, rich	islam, muslim, fundamentalism, radicalized wtc, terrorism, terrorist, terrorist_attacks ground_zero, hijacking, terrorist_targets, security anti_terrorism, anti_americanism, 911, airport

after 5 iterations the stability values do not change considerably. The linear mappings are created using the gradient descent algorithm with a maximum number of 50,000 iterations and a learning rate of 0.01. Before creating embedding spaces, we use the method proposed in [138] to detect bigrams. We consider documents as a combination of unigram and bigram terms.

5.4.3 Intrinsic evaluation

In this section, we describe the dataset we use for evaluating our approaches in detecting semantic shifts.

Ground truth for semantic shifts Following previous work [75, 76, 90, 104, 110], we create a small dataset to evaluate the performance of the proposed approaches. Because we do not possess text-book definitions to evaluate our model—the meaning of the words we study are contested by politicians and academics alike—we assess whether the representations we extract tie in with the perceptions of experts. To validate our method, and see how well we do in the replicating diverging interpretations on political concepts, we choose 24 words which we know were central to many of the controversies of the Thatcher era (1979–1990) and ask experts whether they could recognize the viewpoint. The selected concepts are shown in Table 5.1.

The words we select for evaluation, reflect the prevalent debates of the Thatcher period described in Section 5.4.1, and focus on issues such as economic reform, labour disputes and equality. For each word we select their most similar words in embedding spaces created using speeches give by Conservative and Labour members. Therefore, for each word, we create two lists of similar words. From these lists we discard the overlapping or shared items. These two lists of related terms are then anonymized—meaning that we remove the party where the list stems from—and given to experts, whom we asked if, when shown a concept like 'democracy,' they could identify which list described the Conservative or Labour interpretation. There were 4 annotators who were all political scientists and familiar with the political history of UK. None of the authors participated in the annotation. All annotators annotated all 24 words. The agreement between the annotators, based on Fleiss' Kappa, is 0.47 (p-value$<$0.001) and the overall accuracy is 0.75, indicating that they were able to detect the correct labels in most of the cases. Upon closer inspection, the low agreement may result from the fact that the summaries send mixed signals. The concept 'homosexuality', which was mislabelled by all respondents, is a good example. While the Labour party, at the end of the eighties, was largely supportive of gay rights, the Conservatives took a more negative stance, which led to the infamous Section 28 of the Local Government Act (1988). The phrase 'promoting homosexuality' was as a Labour feature, and could be interpreted as reflecting a more positive opinion, but the same words also figured in the conservative Act, albeit prefixed with 'not'. In general, the summaries fail to capture whether the associated words are in a synonymic or antonymic relation with the target concept, which significantly complicated the interpretation.

5.4.4 Extrinsic evaluation

In this section, we describe the datasets and approaches used for evaluating the proposed word stability measures in document classification and contrastive viewpoint summarization tasks.

Document classification: methods and metrics

We evaluate our stability measures by employing them in the task of ideology detection. The input in this task is a speech held in the UK parliament and the task is to determine the party (the ideology) of the speaker. We train an SVM classifier on a collection of speeches, categorized as either Labour or Conservative. We aim to optimize classifier accuracy by expanding documents as follows: we want to amplify the fact that a speech belongs to a certain class by adding for each unstable word in the speech its top n most similar but unstable words in the embedding space belonging to this class. This is reminiscent of the idea behind doc2vec [121], with the difference that we explicitly change the document. Note that we only expand the documents in the training set, not in the test set. This setup has two parameters. The first is the threshold θ which categorizes all terms as either 'stable' or 'unstable' depending on their stability value. We will optimize this parameter in our experiments using a development set. The second parameter is the number of terms to add for each unstable word. As usual in expansion setups adding too few and too much will lead to worse performance. The effect of n on the performance of different classifiers is shown in Figure 5.5.

We discard speeches of less than 50 words and then randomly select 50,000 speeches from the Thatcher period for each of the parties for performing classification. The mean and median length of the selected speeches are 282 and 107 words, respectively.

We do 10-fold cross validation and report Precision, Recall, and F1 measures in the classification task. We use 8 folds as training data, one fold as development set to tune θ, and one fold for testing. For each document we construct a feature vector using TF-IDF values. Each element of this vector corresponds to a word and its value is TF-IDF weight of the word in the document normalized by the length of the document. After expanding documents in the training set, we re-compute TF-IDF values for words.

As a baseline, we compare the performance against a different expansion method, which inserts neighboring words calculated from an embedding trained on the whole corpus excluding test documents. Since we do not have stability values for this method, we expand all words in speeches using the general embedding. We refer to this method as 'SVM+General' in Section 5.5. The general word embedding is trained on all speeches from the Thatcher period. We compared 'raw SVM' to 'SVM+General' and the latter performed better. Therefore, we take 'SVM+General' as our baseline.

Contrastive Viewpoint summarization: dataset, methods, and metrics

In this section, we describe the evaluation set and our approach for evaluating word stability measures in the contrastive viewpoint summarization task.

Contrastive viewpoint summarization We use the estimated stability values to summarize viewpoints about concepts. The input is a concept w, the stability values estimated using the approaches introduced in this section and the length of the summary l. The output is two lists of summaries in which each list contains l words describing a viewpoint about w.

To summarize a viewpoint V^i about a given concept w, we take the top 100 most similar words to w in embedding space V^i. Then starting from the most similar neighbor, the word is added to the summary if a neighbor is in the overlapping vocabulary of the two embedding spaces and if the stability of the neighbor is equal to or below the set threshold. This process is continued and l words are selected as the summary. As the top 100 neighbors are ordered from highest similarity to lowest, the summaries will follow the same trend. In this task, we set the length of the summary to 5 and the number of iterations of the Neighbor-based method and the Combination method to one (i.e. we only use direct neighbors).

Ground truth for viewpoint summarizarion The summaries produced using the three summarization methods are assessed through peer evaluation. To make the evaluation set, we use the New York Times dataset. We study which method best summarizes the shift in meaning after 9/11. We select a total of 16 concepts, which are chosen based on relevant literature. The selected concepts are shown in Table 5.1. Summarisation questions consist of a concept and its accompanying summaries before and after 9/11 for all three summarisation methods. For each concept, the question is as follows: *In your opinion, which of the summaries belong to the given concept 'Before 9/11' and which of the summaries belong to 'After 9/11'?* No specifications regarding how many summarizations per category were given per concept, leading to a fairly open evaluation. Questions were randomized per survey, as were the options for the summarisation questions. Each summary was annotated by 10 people. The agreement between the annotators, based on Fleiss' Kappa, is 0.54 (p-value$<$0.001). Before asking annotators, we have the labels of the summaries (before 9/11 and after 9/11). A perfect summary would be the one that is annotated by 10 people correctly. Therefore, the number of times the label of a summary generated by a particular method is annotated correctly shows the performance of the method in summarization task. In Section 5.5 we report the performance of different methods as the number of times the annotators detected the labels correctly in terms of Precision, Recall, and F1 of the annotators on the generated summaries.

We use the New York Times dataset instead of the parliamentary proceedings from the Thatcher period for the summarization task. The reason is that it is more straightforward for our annotators to assess whether the summary of a viewpoint about a word belongs to before or after the 9/11 event, compared to assessing whether the summary belongs to the Conservative or Labour party.

5.4.5 Statistical significance

For statistical significance testing, we compare our methods to baselines using paired two-tailed t-tests with Holm-Bonferroni correction for multiple hypothesis testing. We set α (the desired significance) to 0.05. In Section 5.5, $^{\blacktriangle}$ and $^{\blacktriangledown}$ indicate that the corresponding method performs significantly better and worse than the corresponding baseline, respectively.

5.5 Results

In this section, following the four research questions described in Section 5.4, we report the results of different word stability measures.

5.5.1 Results of word stability measures in detecting semantic shifts

To answer **RQ4.1**, we use the dataset described in Section 5.4.3. This dataset contains 24 words which are expected to exhibit ideological divergence. The setup of this experiment can be found in Section 5.4.3. We rank all words in the vocabulary based on the reverse of their stability values (unstable words are ranked higher). A good stability measure should rank the selected words higher. The average rank of the selected words in the ranking created using the Combination method is 462. Based on a paired two-tail t-test, this value is significantly lower than the one for the linear mapping method which is 681. This shows that the proposed approach is effective in finding the words which have different meanings in the UK parliament. Figure 5.2 shows the delta between the rank of the selected words in the rankings created by the two other methods and the Combination method. As can be seen, most of the words are ranked higher by the Combination method compared to the other methods.

We run an additional analysis to see if our methods are robust with respect to semantically stable words. Specifically, we assess if our approaches can detect words that do not change when moving from one party to another. For comparison, we also compute scores using speeches from the Blair period (1997–2007) and compare the tail of the ranking with the tail of the ranking created on the speeches in the Thatcher period. The main intuition is that if a word is stable, its meaning should not change over time (across different periods of the parliament). Figure 5.3 shows the Jaccard similarity of the tails of the two rankings for various tail sizes across all methods. By increasing the size of the tail, more words are included and the intersection of the two lists and the Jaccard similarities are increasing. As can be seen, the Combination method has higher Jaccard similarity values in most of the cases. The Jaccard similarity of the tails when we set the tail size to 5000 (the size of the intersection of 'Labour' and 'Conservative' vocabularies is about 50,000) for the Combination method is 0.83 which is a high value. This value is 0.78 for the Neighbor-based approach and 0.75 for the linear mapping.

Table 5.2 shows the head and the tail of the rankings of words based on instability values estimated for each of the used approaches. As can be seen, all approaches are good in finding highly stable words (the tails), as the tails of the ranking contain very general words which are expected to show little variation across viewpoints. However,

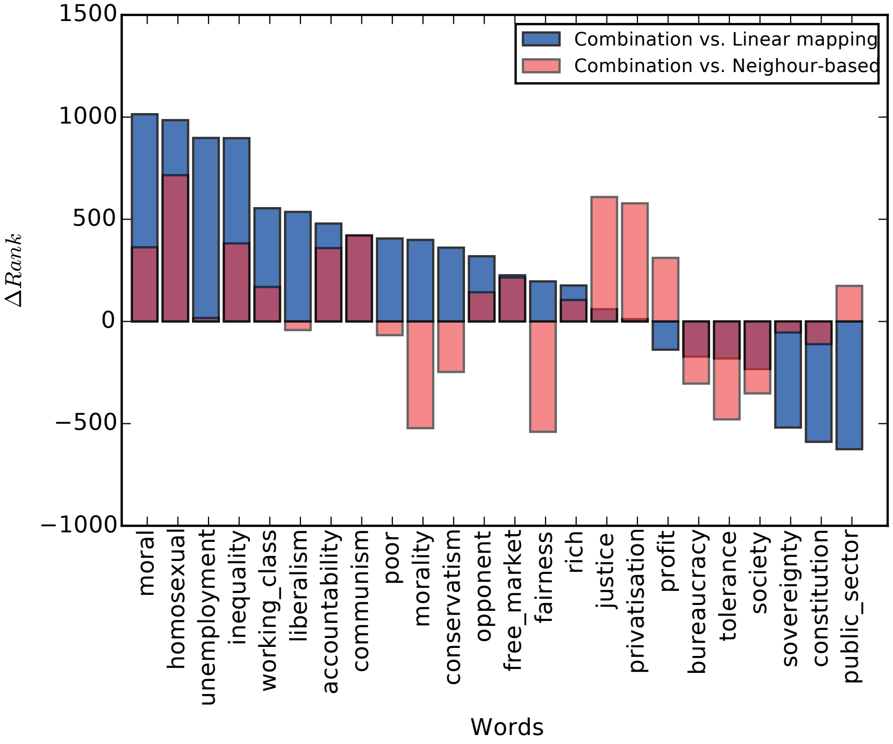


Figure 5.2: The delta between the rank of the selected words in the rankings created by the linear mapping method and the Combination method and the rankings created by the Neighbor-based method and the Combination method.

the head of the list created by the linear mapping approach contains mostly words that we did not expect to shift such as ‘north’ and ‘oil’. Unstable words in the Neighbor-based method’s list such as ‘socialist’ and ‘democratic’ are expected to vary. This method is effective in finding these words. However, there are words such as ‘noble’ and ‘space’ in top of this list. Based on our analysis, the Conservatives included more aristocratic members (which are addressed as ‘noble’ Friend) while Labour MPs use ‘noble’ as a an adjective to reflect the quality. Also, Conservatives use the word ‘space’ when they refer to ‘space technology.’ However, Labour use the word ‘space’ to mostly speak about ‘living space or urban space.’ Therefore, these two words do diverge and the two parties use these words in different contexts to describe different concepts, but the relationship with ideology is not always straightforward.

From the results presented here we conclude that the Combination method is highly effective in detecting semantic shifts (as shown in Figure 5.2) and very robust with respect to semantically stable words (as shown in Figure 5.2 and Table 5.2).

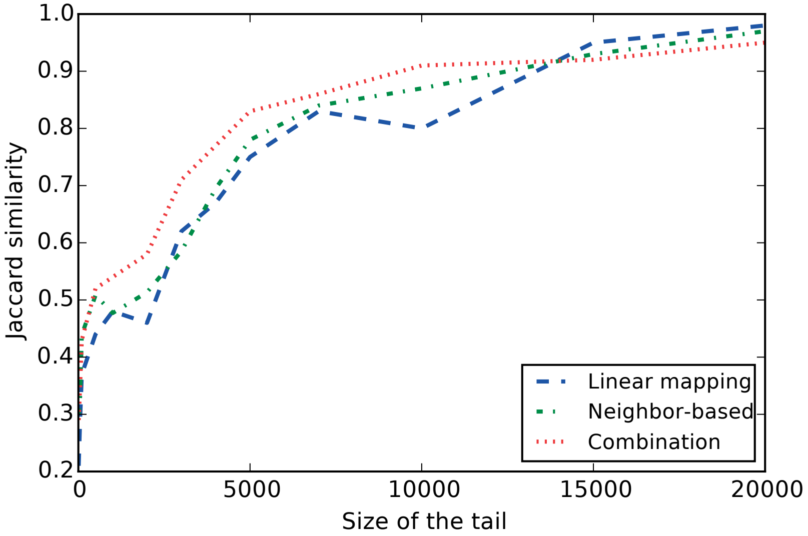


Figure 5.3: Jaccard similarity of tails of the rankings created for the Thatcher and the Blair period using linear mapping, Neighbor-based, and Combination methods.

5.5.2 Results of word stability measures in document classification

To answer **RQ4.2**, we use the method described in Section 5.4.4 for expanding speeches in the UK parliament during the Thatcher period and employ the expanded documents for classifying speeches by party. The setup of this experiment can be found in Section 5.4.4.

Table 5.3 shows the results of this experiment. In general, the results indicate that the proposed word stability measures help in discriminating documents. Moreover, two other observations can be made from the results. First, expanding documents, even with a general embedding, can improve performance of the classifiers. Second, the Combination method performs better than the other approaches. The linear mapping approach does not outperform the baseline. The higher accuracy of the Combination method shows that, although the linear mapping approach does not improve the performance of the classifier, when it is combined with the Neighbor-based method, the performance is improved.

To gain additional insights about our approaches, we further analyze speeches that are correctly classified by the Combination method but not by the "SVM+General". The following (part of a) speech is an example of such samples:

> "...subsidise the residents of wasteful *labour* authorities. If we were to strip away the surcharges and handouts, we would find that the *labour* party's arithmetical inexactitude is almost a case for reference to the advertising standards authority. Having done that, we find that totally conservative areas have an average community charge of 305 pound, compared with the rip-off in totally *labour* areas of 412 pound. Opposition members may think that this is a laughing matter, but a differential of no less than 107 pound per head for the privilege of voting *labour* has a devilish

Table 5.2: The head and the tail of ranking of words achieved using different word stability measures. For the Neighbor-based and the Combination methods the number of iterations is set to 5.

Method	Head	Tail
Linear mapping	gas nuclear_power north oil church	member tuesday thursday thank nothing
Neighbor-based	noble socialist illegal democratic space	wednesday friday monday tuesday december
Combination	legislative inequality private_enterprise noble democratic	about tuesday side nothing thursday

Table 5.3: Results of classification of speeches to parties using different word stability measures. We consider SVM+General as our baseline.

Method	Precision	Recall	F1
SVM	0.781	0.718	0.758
SVM+General	0.793	0.735	0.763
SVM+Linear mapping	0.804▲	0.738	0.770
SVM+Neighbor-based	0.823▲	0.768▲	0.795▲
SVM+Combination	0.839▲	0.775▲	0.806▲

> impact on the charge payers of those areas.”

This speech was given by a member of the Conservative party. However, it is mostly about the Labour party since it mentions the ‘Labour’ party more than 4 times. ‘Labour’ is a very discriminative word for Labour party and that is the main reason that this speech is classified in the Labour class. However, when we expand documents in the training step with words the Conservatives characteristically use to describe Labour members, the different sense of the word becomes apparent. Thus, in the example mentioned above, the words such as ‘subsidize’, ‘wasteful’ and ‘inexactitude’ will help more to classifying this example correctly.

Moreover, we analyze the speeches that are classified correctly by ‘SVM+General’ and incorrectly using the stability measure. Our analysis show that most of these speeches are very short ones that do not contain any information about the author’s viewpoint. When we filter out documents with less than 200 words length, the F1 score of ‘SVM+General’ is increased to 0.79 and the F1 score of the Combination method is

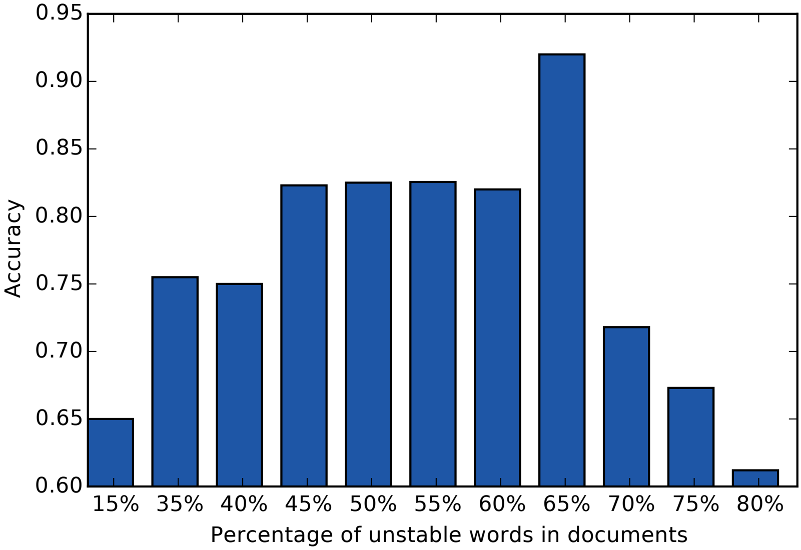


Figure 5.4: The accuracy of the Combination method in classifying the speeches in the Thatcher period for different levels of expansion. $\%i$ in x-axis is representing the documents that $\%i$ of their words are unstable.

0.85 and the improvement of the Combination method is more than the improvement of ‘SVM+General’. Another source of error is inaccurate stability values for words. This causes the expansion of documents with wrong words and lowers the accuracy of the classifier. Figure 5.4 shows the accuracy achieved for different levels of stability values. We first calculate the percentage of unstable words (words for which their stability calculated using the Combination method is less than θ) in documents. Then, we put the documents into different bins based on their percentage of unstable words and calculate the accuracy of the classifier for each bin. We only show the bins containing more than 1000 documents. For the highly unstable documents the accuracy is the lowest. This is mostly due to extreme expansion of these documents (since their words have low stability values) which are not accurate enough in most of the cases. The accuracy is higher when the stability value does not skew towards one of the extremes.

5.5.3 Results of word stability approaches in contrastive viewpoint summarization

This section answers **RQ4.3**. We use different word stability measures and the method described in Section 5.4.4 to generate summaries for words. The setup of this experiment can be found in Section 5.4.4. To evaluate the performance of our methods in summarizing the viewpoints, we use the dataset described in Section 5.4.4 and report results in Table 5.4. In general, the performance of the Combination method is slightly better than the linear mapping approach. However, the difference is not statistically significant. The F1 score achieved using the Combination method is 0.75 which is rea-

Table 5.4: Results of different word stability measures in summarizing the viewpoints.

Method	Precision	Recall	F1
Linear mapping	0.74	0.74	0.74
Neighbor-based	0.66	0.64	0.65
Combination	0.75	0.74	0.75

sonably good and indicates that the annotators were able to detect the viewpoints using the provided summaries. The results show that the linear mapping method performs better than the Neighbor-based approach in the summarization task. This result is in contrast with the results achieved in the classification task. The summarization task is done on the New York Times dataset, while the classification task is performed on the UK parliamentary proceedings. In the parliamentary proceedings, the viewpoints are more apparent as neighbors of a given word generally serve as reliable descriptors of the viewpoint. Therefore, the Neighbor-based approach which is solely based on the similarity of the neighbors in two spaces performs better than the linear mapping method in the classification task.

5.5.4 Statistical laws of semantic change

In this section,we answer **RQ4.4**. Recently, Hamilton et al. [75] proposed two quantitative laws of semantic change: (1) the law of Conformity which implies that “the rate of semantic change scales with an inverse power-law of word frequency”. (2) the law of Innovation which reflects that “the semantic change rate of words is highly correlated with their polysemy”. To test if, besides accounting for change over time, these laws also help explain semantic shifts across ideological perspectives, we used the UK parliamentary proceedings from the Thatcher period. To check the first law, we construct two vectors (each entry in these vectors corresponds to a word and the length of the vectors is equal to the size of the intersection of vocabularies of Labour and Conservative parties): one using the frequency of words and one using their instability ($1 - stability$). Then, we calculate the Pearson correlation between these vectors. To check the second law, again we construct two vectors in a similar way: one using the polysemy of words (we use WordNet to calculate the number of senses of words to quantify their polysemy) and one using the instability of words.

The Pearson correlation values are shown in Table 5.5. The results show that: first, the law of conformity strongly holds using all measures. This becomes even more apparent when we use the linear mapping method. This is expected since we use highly frequent words for training the mappings. Second, the law of innovation does not strongly hold using all measures. We hypothesize that this is because even when parties use a word with low polysemy, they inject it with diverging meanings for example by using different sentiment words to express their opinion about the word.

Moreover, we hypothesize that lexically different word senses are unlikely to appear in a short period, or in (still very similar) the political data we use. Thus, there are likely other, deeper causes such as concreteness of words. We study how the semantic change rate is correlated with the concreteness of words. Again, we construct two vectors: one using the concreteness of words and one using their instability. We use a dataset [31] containing the concreteness rating of words for constructing the concreteness vector. The results are shown in Table 5.5. The result indicate that there is a negative correlation between concreteness and instability and concrete words are less likely to shift. In fact, more abstract words are more likely to shift.

Table 5.5: The Pearson correlation between the instability of words with their frequency, polysemy, and concreteness.

Measure	Conformity	Innovation	Concreteness
Linear mapping	-0.63	0.11	-0.31
Neighbor-based	-0.42	0.18	-0.34
Combination	-0.51	0.22	-0.39

5.5.5 Analysis

In this section, we analyze the quality of the trained mappings between the embedding spaces and measure the effect of the word-expansion on the classifier's accuracy.

Quality of linear mappings

To measure the quality of the created mappings, we report the average value estimated using Equation 5.3 over all words in the vocabulary. Table 5.6 shows the results of this experiment. The average similarity calculated using one-way mapping ($sim_{ij}(w) = \cos(W^{ij}V^i_w, V^j_w)$) is low, meaning that when words are mapped from one space to the other, they are not close to the same word in the destination space. However, when we use Equation 5.3, the average similarity value is high indicating that when going back to the same space, the word is mapped to its original vector. This shows that a low value of similarity for the one-way mapping is mainly due to instability of words for which their location in two spaces (and hence their meaning) are different based on different viewpoints.

Table 5.6: Average cosine similarities of words after linear mappings on the UK parliamentary proceedings in the Thatcher period and the New York Times datasets. i and j are the source and destination spaces of the mappings. Con and Lab are Conservative and Labour. Before and after are embeddings created for before 9/11 and after 9/11.

Dataset	Setting	$\cos(W^{ij}V^i_w, V^j_w)$	$\cos(W^{ji}W^{ij}V^i_w, V^i_w)$
UK	$i = Con, j = Lab$	0.43	0.84
	$i = Lab, j = Con$	0.43	0.85
NY Times	$i = before, j = after$	0.49	0.85
	$i = after, j = before$	0.48	0.87

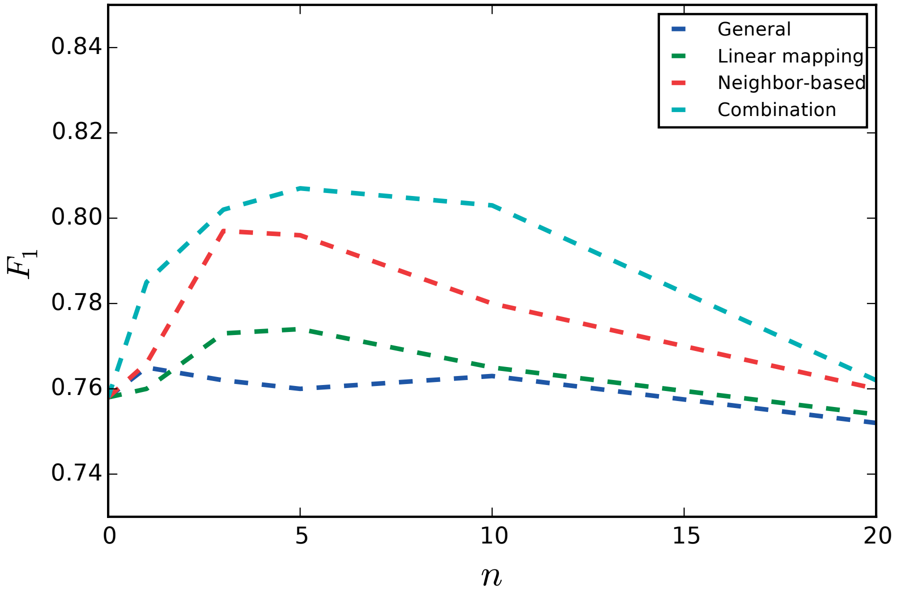


Figure 5.5: The effect of the number of expansion words on the F1 score of speech to party classification using different word stability measures. n is the number of expansion words.

Parameter analysis

In this section, we analyze the effect of the number of expansion words on the effectiveness of word stability approaches in the document classification task. Figure 5.5 shows the F1 scores achieved using different methods based on different number of expansion words. For $1 < n < 3$, the F1 score for all approaches is increased by increasing n. Moreover, for the Combination method even adding 5 expansion words boosts the performance of the classifier. This shows that adding more unstable words to documents can help in discriminating documents belonging to different parties. For $n > 5$, adding more words decreases the performance of the classifiers for all methods and the performance of all approaches are almost the same for $n = 20$. This result indicates that by adding more than a certain number of words, the expanded documents become more and more similar, regardless of the measure used.

5.6 Conclusion

We introduced a general framework for computing semantic shifts by using word embeddings trained on corpora that (are presumed to) represent specific viewpoints. We proposed several methods that compare words across these vector spaces–with their different dimensionality structures–and have demonstrated how these algorithms capture valuable changes in word meaning. We evaluated the results on political speeches and media reports. In doing so, we have shown that the techniques developed here adequately detect words that exhibit ideologically or chronologically diverging sense, *and* can be applied to different types of discourse. We showed that semantic shifts not only occur over time but also across viewpoints.

Our results demonstrated that the proposed word stability measures contribute to other tasks such as contrastive viewpoint summarization, which generates summaries that explicate the diverging viewpoints, and document classification. Moreover, we showed that temporal laws of change also apply to other dimensions. Our results demonstrated that the law of conformity strongly predicts the (in)stability of words, while the law of innovation only has a minimal effect. This indicates that the meaning of frequent words do not shift across viewpoints, while even the meaning of words with low polysemy values can shift. Furthermore, we proposed another law for semantic shifts which implies that more concrete words are insensitive to the viewpoint of speaker.

To the best of our knowledge, this study is the first attempt to detect semantic shifts across viewpoints. We hope that the created datasets and proposed approaches will be beneficial to future research in this area.

The estimated stability measures can be useful in various applications. As shown in this research, it can be used for summarizing diverging viewpoints and document classification. The generated summaries can be used in an exploratory search scenario to uncover diverging aspects of a given topic. In this study, we only focused on detecting shifts in political and media discourse, but our approaches are applicable in any other kind of discourse such as different groups in social media.

Future work will focus on broadening the set of applications, by, for example, examining how our approach contributes to controversy detection and locating people in the "filter bubble." If the language use of the specific group exhibits radically divergent word meanings, then they might be in the filter bubble and word stability can be used to quantify this.

In this chapter, we studied how metadata features can be helpful in an exploratory search task. Next, we will focus on a different exploratory search task and study how structure and metadata can be useful in finding similar questions in community question answering forums.

6 Learning Question Representations for Question Retrieval using Content, Structure and Attention

In this chapter, as the second case study, we focus on the task of similar question retrieval in community questions answering forums. We study how the structure and metadata associated with questions on these forums can be helpful in estimating the similarity of questions and answer our research question **RQ5**.

6.1 Introduction

Community Question Answering (cQA) sites such as Stack Exchange and Yahoo! Answers have become key platforms for exchanging knowledge [46, 142, 225, 231]. Users can post questions on different topics and receive high quality answers from experts. On cQA sites, knowledge is captured in the form of questions from questioners and the associated answers contributed by answerers. Questions and answers on cQA sites are typically highly structured and contain different data fields such as question title, body, category information, answer scores (upvotes) given by users, best answer information, publication dates of questions and answers, etc.

The task of finding questions that are relevant to a newly posted question is called *question retrieval*. Here, a question is relevant to a newly posted question if they have a similar meaning and they are about a similar information need. The task matters for a number of reasons. It can help to detect *duplicate questions*, i.e., questions that are duplicates of questions that have already been answered on the cQA platform, and thereby help reduce answerer effort [142, 178]. And it can help address *question starvation* [46, 142, 152], which occurs when a new question is posted and there is no immediate answer provided by answerers for it: if there is an already answered relevant question, the answers to this question may provide an answer to the new question.

The main focus of this study is on question retrieval on cQA sites. This task is challenging for two main reasons. The first has been well studied; it is the *vocabulary gap* [46, 152, 225, 231]: the phenomenon that different users express the same question

This chapter was published as [11].

using different words. The second problem has been less well studied; it is what we call *expressive inefficiency*, the phenomenon that users do not explain their questions with enough details.

Previous approaches to question retrieval have met with mixed success in addressing the two challenges, the vocabulary gap and expressive inefficiency. One category of question retrieval methods consists of traditional lexical matching methods [96, 139, 211, 229]. These approaches suffer from the vocabulary gap. Semantic matching models try to overcome the vocabulary gap by modeling the semantics of questions [142, 152]. Topic model-based approaches try to bridge the vocabulary gap by modeling topical similarity of questions [32, 93]. And recently, neural network-based approaches have been used to learn semantic representations of questions that can be used to estimate the similarity of questions [46, 122, 142, 149, 231]. These approaches achieve state-of-the-art performance on the question retrieval task [142]. To address expressive inefficiency previous work has tried to use question-answer pairs instead of question-question pairs to learn a semantic matching model [46]. The idea is that since answers contain more information and details about the questions they are answering, learning a retrieval model based on answers can help overcome expressive inefficiency.

Most neural network-based question matching models only use the title of questions for learning representations of questions, neglecting other valuable information such as answers given to questions and tags associated with questions. When a new question is posted, it usually contains a title, a body, and tags associated with it. However, for existing questions much more information is available such as answers, comments and scores. Our working hypothesis is that such information and structure can be used to address both the vocabulary gap (in a better way than semantic matching models that only use the title or body of questions) and expressive inefficiency challenges. It is not straightforward to use multiple data fields in a unified way, de-noise the fields, and match questions based on the representations achieved from the de-noised fields.

In this work, we propose a neural method for question retrieval that is effective in addressing both the vocabulary gap and expressive inefficiency. Our proposed method addresses the vocabulary gap by using the rich set of information available on cQA platforms, not just the titles of questions, like most work which also tries to address the the vocabulary gap before this study, but also the structure and data fields associated with questions, such as tags and best answer information. Our proposed method addresses expressive inefficiency by using a novel multi-context attention mechanism that allows it to use different fields of questions differently, weight them, de-noise their representations, and select the most relevant fields to be matched against indexed questions and answers.

More specifically, we propose a neural architecture, called *Question Retrieval using Content, Structure and Attention* (CSA), that first learns different representations for different fields of a newly posted question Q_q. For textual fields such as a question's title and body, we take a hierarchical approach. We first divide the input into sentences and create representations of sentences by taking the average of embeddings of their words. Then, we consider each textual field as a sequence of sentences. We use a bi-directional RNN to produce the representation of each field using the representations of its sentences. We treat tags as sequences of words and use a bi-directional RNN to produce a representation of tags. For indexed questions Q_c (including their answers and tags), we use a similar approach. We learn representations of textual fields by taking the average of words in sentences and considering the input as a sequence of sentences. After learning representations for different fields of questions, we use the representations learned for Q_q as a context and employ a multi-context attention mechanism to weight different fields of Q_c. Finally, we use the weighted representation of Q_c to predict the relevance label of Q_c.

Our main goal in this study is to find out how we can use multiple data fields associated with questions to find relevant questions to a newly posted question on cQA sites and how different data fields can be incorporated to retrieve questions more accurately. To this end, we address **RQ5**:

> *Does using the rich data and structure available on question answering forums lead to a better score than the state of the art on the task of question retrieval?*

We evaluate the proposed approach, CSA, using a Stack Overflow dataset[1] to train the model and select a subset of this dataset for evaluation. To the best of our knowledge, this is the only large and publicly available dataset in which questions have multiple fields and in which there are enough labeled samples. Most existing datasets, such as the Yahoo! Answers and SemEval [142] datasets, have tags and answers associated with questions. However, these datasets have a very small set of labeled data (pairs of questions with relevance labels), making it hard to use them to train an effective state-of-the-art neural model that exploits multiple fields.

Our main contributions are: (1) Question Retrieval using Content, Structure and Attention (CSA), a model that automatically learns representations of questions submitted to cQA sites based on different fields and structure associated with them. Most existing question retrieval methods only use the question text and best answer. CSA jointly models and uses multiple data fields associated with questions. (2) A multi-context attention mechanism that can use multiple signals to weight different fields of questions. Most of the fields associated with questions provide valuable information to retrieve questions. We perform extensive analyses to assess the importance of different fields of questions for learning the question representations.

6.2 Problem Definition

Given a question Q_q and an archive of previously submitted (and answered) questions, we need to rank all archived questions according to their relevance to Q_q. An archived question is said to be relevant to the new question Q_q if both are about the same topic and seek answers to similar information needs. More precisely, the input in the question retrieval task is a question Q_q and a set of archived questions $Arch = \{Q_c^1, Q_c^2, \ldots\}$. The output is a ranking of questions in $Arch$ based on their relevance to Q_q.

We assume that questions have multiple fields. Each question is represented as a vector consisting of four fields: title, body, tags, and best answer. For Q_q the best answer field is empty. For archived questions, the best answer field is assumed to be

[1] The Stack Overflow dataset can be obtained from `https://archive.org/download/stackexchange`. We use the following files: stackoverflow.com-Comments, stackoverflow.com-PostHistory, stackoverflow.com-PostLinks, stackoverflow.com-Posts, and stackoverflow.com-Tags.

non-empty. We use these specific fields since they are available in almost every cQA platforms and our goal is to make the model broadly applicable. Each textual field (title, body, and best answer) is represented as a sequence of words (unigrams) within sequences of sentences. Tags are represented as sequences of unigrams.

In order to rank questions in $Arch$, we take a point-wise approach and first estimate the similarity of each question in $Arch$ to Q_q independently and then rank the questions based on the similarity scores. Our proposed neural network has two inputs: Q_q and Q_c^i (a question from $Arch$) and an output: a similarity score.

6.3 CSA: Question Retrieval using Content, Structure and Attention

We first introduce the architecture of the proposed neural network. Then, we describe the details of our input representation, employed attention mechanism, and the training objective.

6.3.1 Model overview

The setting is as follows. We are given a question Q_q, the query question, for which n data fields, like title, body text, and associated tags are available, i.e., $\{f_1^q, f_2^q, \dots, f_n^q\}$, and a set of candidate question, $\{Q_c^1, Q_c^2, \dots\}$, each of which have m data fields like title, body text, associated tags, and best answer, i.e., $\{f_1^c, f_2^c, \dots, f_m^c\}$.[2] Our task is to rank the candidate queries based on their probability of relevance to the query question. Our proposed model takes all the data fields of the query question and the data fields associated with one of the candidate questions and predicts the probability of the candidate question to be relevant to the query question. See Figure 6.1. The model contains three major building blocks:

Input representation module that provides a set of functions for encoding different data fields of the input, either from the query question or candidate question, each to a dense vector x_i^qor $x_i^c \in \mathbb{R}^d$. For simplicity, we assume that d is the dimension of the representation vector of all data fields. These functions are defined to encode different types of the data field with different properties. See Section 6.3.3.

Multi-context attention module where an attention mechanism is employed to address the expressive inefficiency problem by weighting and exploiting the representation of data fields of the candidate question, i.e., $\{x_i^c\}_{i=1}^m$, given the representation of different data fields of the query question, i.e., $\{x_i^q\}_{i=1}^n$. The output of this module is two vectors: $s^c \in \mathbb{R}^d$ and $s^q \in \mathbb{R}^d$, which can be considered as the aggregated de-noised representations of the candidate question and the aggregated representation of the query question, respectively. See Section 6.3.2.

[2] To simplify our notation, we drop the index of the candidate from its fields' notation as the proposed model is a point-wise model, dealing with a single candidate at a time.

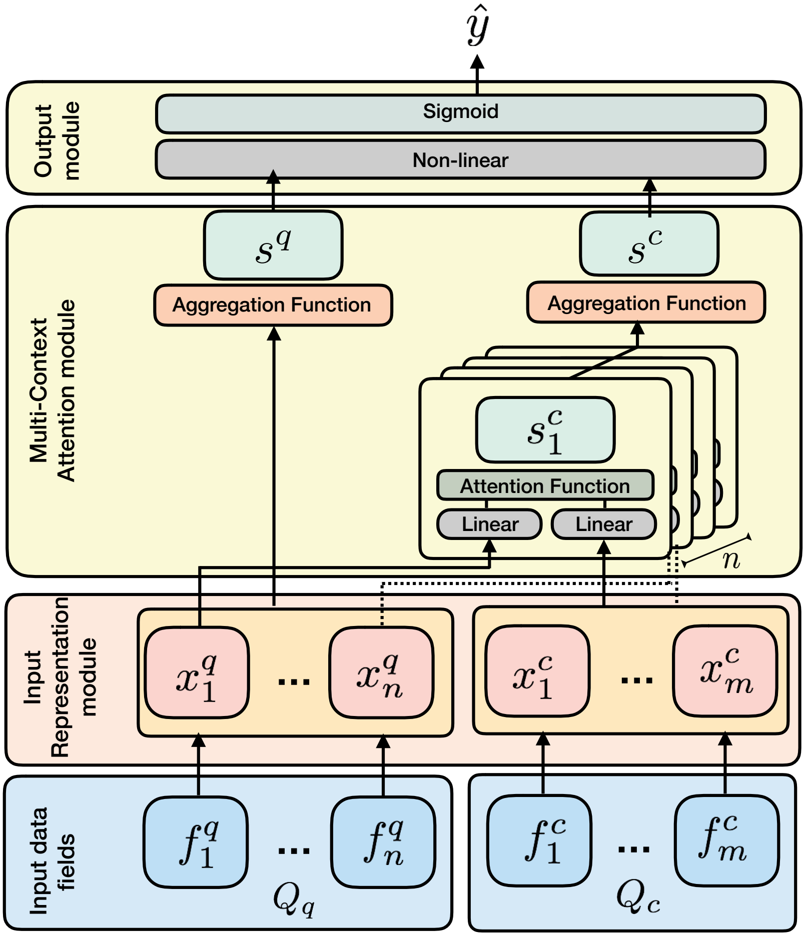


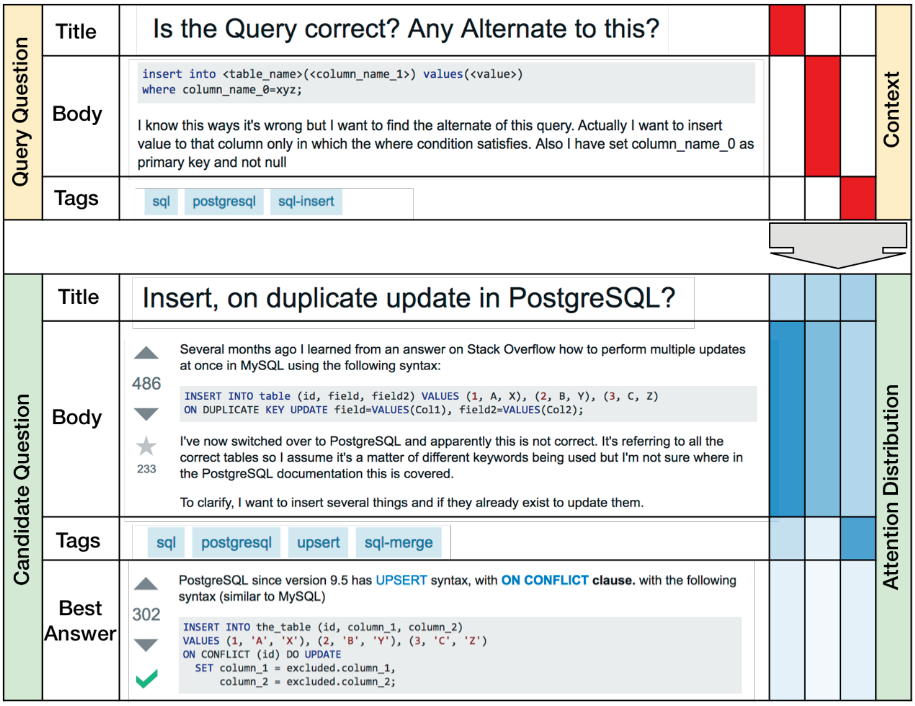
Figure 6.1: General schema of the proposed model, CSA.

Output module The output module captures the interaction between the query question representation, s^q, and the candidate question representation, s^c, that are provided by the previous module and predicts the probability of their relevance $\hat{y}$. See Section 6.3.4.

Below, we first define the multi-context attention unit, and then we explain the input representation and output modules and how these modules are connected to each other.

6.3.2 Multi-context attention

We hypothesize that using multiple data fields for the candidate question can address the expressive inefficiency problem. It is not straightforward to aggregate all fields in a unified way, in particular for learning an effective representation for the (archived) candidate question Q_c. To tackle this problem, we propose an attention mechanism to dynamically aggregate information from different data fields of Q_c, with respect to



Is the Query correct? Any Alternate to this?

```
insert into <table_name>(<column_name_1>) values(<value>)
where column_name_0=xyz;
```

I know this ways it's wrong but I want to find the alternate of this query. Actually I want to insert value to that column only in which the where condition satisfies. Also I have set column_name_0 as primary key and not null

Insert, on duplicate update in PostgreSQL?

Several months ago I learned from an answer on Stack Overflow how to perform multiple updates at once in MySQL using the following syntax:

```
INSERT INTO table (id, field, field2) VALUES (1, A, X), (2, B, Y), (3, C, Z)
ON DUPLICATE KEY UPDATE field=VALUES(Col1), field2=VALUES(Col2);
```

I've now switched over to PostgreSQL and apparently this is not correct. It's referring to all the correct tables so I assume it's a matter of different keywords being used but I'm not sure where in the PostgreSQL documentation this is covered.

To clarify, I want to insert several things and if they already exist to update them.

PostgreSQL since version 9.5 has UPSERT syntax, with **ON CONFLICT clause.** with the following syntax (similar to MySQL)

```
INSERT INTO the_table (id, column_1, column_2)
VALUES (1, 'A', 'X'), (2, 'B', 'Y'), (3, 'C', 'Z')
ON CONFLICT (id) DO UPDATE
  SET column_1 = excluded.column_1,
      column_2 = excluded.column_2;
```

Figure 6.2: Example of attention distribution given different contexts. In the top half of the figure, a query question with three fields is shown. The red color indicates which field of the query questions is used as the context vector. In the bottom half of the figure, a candidate question with four fields is shown. The blue color shows, for the corresponding context vector, the distribution of attention weights on the fields of candidate question.

the given data fields of query question Q_q. The original attention function [192, 204] can be described as mapping a context (also known as query) and a set of key-value vector pairs to an output vector, where the output vector is supposed to be an effective "summary" of the values focusing on information linked to the given context. In our setting, we have multiple contexts and we propose an attention mechanism to learn a de-noised representation for candidate questions. The output is computed as a weighted sum of the values, and the weights assigned to different values (attention distribution), are computed by a compatibility function of the context vector with the corresponding key vectors.

For simplicity, we assume that each vector x_i^c serves both as the key to compute the attention vector over data fields of candidate questions Q_c and as the value to encode the representation of the i-th field of the candidate question. However, in our setup, we have multiple context vectors, i.e., data fields of Q_q, and each might lead to a different attention distribution. Figure 6.2 shows a heat map representing the distribution of attention given different contexts for a real example. As can be seen, given the title of the query question as the context vector, the compatibility function assigns a high weight to the value vector representing the body text of the candidate question. The value vector representing the tags associated with the candidate gets the highest weight in the attention distribution when the vector representing tags of the query question is considered as the context vector.

To capture the importance of different value vectors given multiple context vectors, we propose a *multi-context attention mechanism*. Multi-context attention consists of n attention functions, each taking $x_i^q \in \mathbb{R}^d$, i.e., the vector representing the i-th data field of the query question as the context vector to estimate the contribution of each value vector to a final local summary vector s^i. Given $X^c = [x_1^c, x_2^c, \dots, x_m^c] \in \mathbb{R}^{d \times m}$ as the concatenation of all value vectors, we compute the local summary vector s_i^c as follows:

$$
\begin{aligned}
Z_i &= \tanh(\boldsymbol{W}^{(c)} X^c + (\boldsymbol{W}^{(q)} x_i^q)\mathbf{1}^T) && \in \mathbb{R}^{d \times m} \\
\alpha_i &= \text{softmax}(\boldsymbol{w}^T Z_i) && \in \mathbb{R}^{1 \times m} \\
s_i^c &= X^c \alpha_i^T && \in \mathbb{R}^d,
\end{aligned}
$$

where $\mathbf{1} \in \mathbb{R}^m$ is a vector of ones, $\boldsymbol{W}^{(q)}, \boldsymbol{W}^{(c)} \in \mathbb{R}^{m \times m}$ are trainable projection matrices, and $\boldsymbol{w} \in \mathbb{R}^m$ is a trainable vector.

The final summary of the candidate question, denoted by $s^c \in \mathbb{R}^d$, is generated by an aggregation function, a feed-forward layer to compute a non-linear combination of all the local summaries. Assuming $S^c = [s_1^c : s_2^c : \dots : s_n^c] \in \mathbb{R}^{dn \times 1}$ is the concatenation of all local summaries generated by all the attention functions, we have:

$$
s^c = \tanh(\boldsymbol{W}^{(I_1)} S^c + \boldsymbol{b}^{(I_1)}) \qquad \in \mathbb{R}^d,
$$

where $\boldsymbol{W}^{(I_1)} \in \mathbb{R}^{d \times dn}$ and $\boldsymbol{b}^{(I_1)} \in \mathbb{R}^d$ are a trainable projection matrix and bias, respectively.

Similarly, having $X^q = [x_1^q : x_2^q : \dots : x_n^q] \in \mathbb{R}^{dn \times 1}$ as the concatenation of all context vectors, we apply a similar aggregation function to generate a summary vector for the query question:

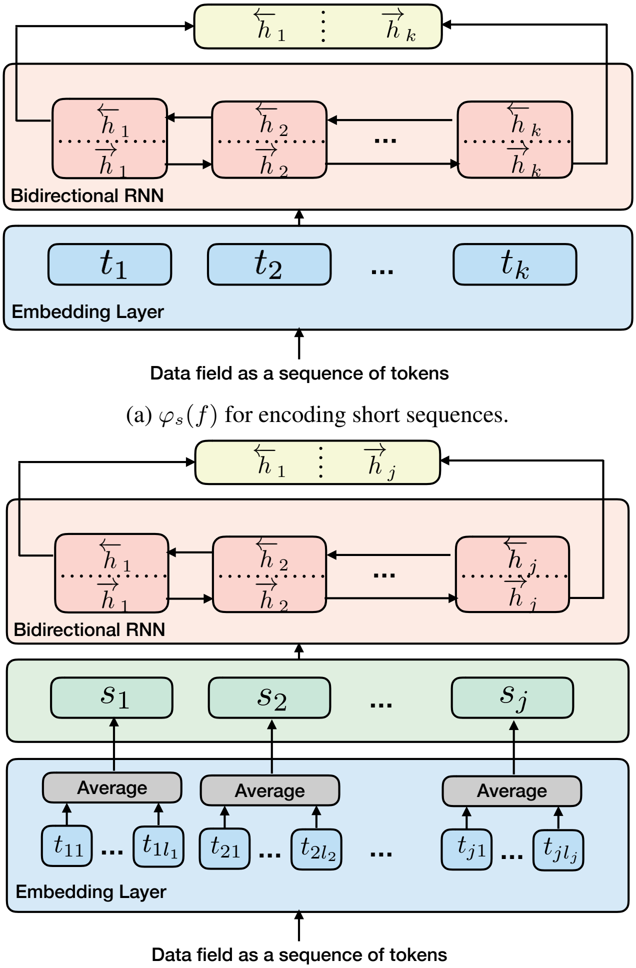
$$
s^q = \tanh(\boldsymbol{W}^{(I_2)} X^q + \boldsymbol{b}^{(2)}) \qquad \in \mathbb{R}^d,
$$

where $\boldsymbol{W}^{(I_2)} \in \mathbb{R}^{d \times dn}$ and $\boldsymbol{b}^{(I_2)} \in \mathbb{R}^d$ again are a trainable projection matrix and bias, respectively.

6.3.3 Input representation

The input representation layer of CSA consists of a set of encoding functions that are in charge of mapping data fields, both from the query question and the candidate question to a set of dense vectors with dimension d. Defining these encoding functions depends on the type and properties of the available data fields. We assume that we have a title, a body text, and a sequence of tags for the query question and a title, a body text, a sequence of tags, and the text of the best answer for the candidate question.

With respect to the properties of these data fields, we have tried different encoding functions and ended up with two main functions $\varphi_s(T)$ and $\varphi_l(T)$, where the first one is for encoding short sequences of words and the second one is specialized for encoding



(a) $\varphi_s(f)$ for encoding short sequences.

(b) $\varphi_l(f)$ for encoding long sequences.

Figure 6.3: Architectures of encoding functions in the input representation module.

longer sequences of words.[3] In both, we first have a trainable embedding layer that maps words into vectors of dimension d. Let us now explain these two functions.

In CSA, $\varphi_s(T)$ is simply a bidirectional recurrent neural network (RNN) that reads the input sequence $T = [t_1, t_2, \ldots, t_k]$ in the left-to-right direction in the RNN forward pass. It creates sequences of hidden states, $[\overrightarrow{h}_1, \overrightarrow{h}_2, \ldots, \overrightarrow{h}_k]$, where $\overrightarrow{h}_i = \text{RNN}(t_i, \overrightarrow{h}_{i-1})$ is a dynamic function for which we can use, for example, an LSTM [83] or a GRU [38]. The RNN backward pass reads T in the reverse direction, i.e., $\overleftarrow{h}_i = \text{RNN}(x_i, \overleftarrow{h}_{i+1})$, resulting in a sequence of hidden states $[\overleftarrow{h}_k, \overleftarrow{h}_{k-1}, \ldots, \overleftarrow{h}_1]$.

We take the concatenation of the last hidden state of the forward pass and the first hidden state of the backward pass of the RRN, i.e., $\varphi_s(T) = [\overrightarrow{h}_k; \overleftarrow{h}_1]$, as the final

[3] We use tokens and words interchangeably to represent unigrams.

representation for the given data field (see Figure 6.3a).

To learn a representation for longer sequences, similar to [103, 215] we take a hierarchical approach. We define another encoding function, $\varphi_l(T)$, in which first we extract sentences from the input text and represent the input as a sequence of sentences, and then, considering each sentence as a bag of words, we average word vectors to compute sentence vectors. We consider these sentence vectors as a sequence of tokens and use a bidirectional RNN (similar to the one we employed in $\varphi_s(T)$) to encode the given data field to a vector (see Figure 6.3b). We encode tags with $\varphi_s(f)$ and the other data fields with $\varphi_l(f)$. Although titles are usually short, we choose to encode titles using $\varphi_l(f)$ as based on our experiments there are some quite long titles which are hard to encode using $\varphi_s(f)$. Tags associated with a question are treated as a sequence, and not as a set, of tokens, to be able to capture the order of tags in the representation. Our hypothesis is that when choosing tags, users pick the most important tags first and there is an implicit order between chosen tags. We try to model this order using an RNN.

6.3.4 Output module

The output module of CSA, which is in charge of capturing the interaction between query and candidate question, is a feed-forward layer with tanh non-linearity, followed by a sigmoid. It receives the summary vectors of the query question and the candidate question and predicts the probability of them being similar:

$$\begin{aligned} O &= \tanh(\boldsymbol{W}^{(O)}[s^q : s^c] + \boldsymbol{b}^{(O)}) && \in \mathbb{R}^{2d} \\ \hat{y} &= \text{sigmoid}(\boldsymbol{w}^T O) && \in \mathbb{R}, \end{aligned}$$

where $\boldsymbol{W}^{(O)} \in \mathbb{R}^{2d \times 2d}$ and $\boldsymbol{b}^{(O)} \in \mathbb{R}^{2d}$ are a trainable projection matrix and bias, and $\boldsymbol{w} \in \mathbb{R}^{2d}$ is a trainable vector.

6.3.5 Training objective

We train CSA in an end-to-end training process in which, given mini-batches of data $\langle Q_q, Q_c, y \rangle$, we first predict $\hat{y}$ as the estimated probability of the candidate question being relevant to the query question and calculate the loss using the cross entropy loss:

$$L(Q_q, Q_c) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}). \tag{6.1}$$

6.4 Experimental Setup

We aim to understand how we can use all information associated with questions to find relevant questions to a newly posted question in cQA forums and how different data fields can be incorporated to retrieve questions more accurately. To this end, we answer our fifth research question:

RQ5 Does using the rich data and structure available on question answering forums lead to a better score than the state of the art on the task of question retrieval?

RQ5 is concerned with the impact of using different data fields and the structure associated with questions on the performance of question retrieval methods. To answer this question, we use a Stack Overflow dataset to evaluate the performance of CSA and compare it to the performance of the state-of-the-art for the question retrieval task. Moreover, to gain additional insights on how CSA and multi-context attention mechanism work, we perform several analyses and answer the following research questions:

RQ5.1 Does the attention mechanism help to retrieve relevant questions more accurately?

RQ5.2 What are the most informative components of questions for estimating the similarity of questions?

RQ5.3 Is it beneficial to combine CSA with previous approaches?

To answer **RQ5.1**, we study the effect of using an attention mechanism in question retrieval. We compare the performance of CSA to a variation of it without the attention mechanism.

RQ5.2 concerns the effectiveness of the proposed attention mechanism in the question retrieval task. To answer **RQ5.2**, we analyze the weights given to different parts of questions by the attention mechanism.

RQ5.3 is concerned with the impact of combining different question retrieval methods on the performance on the question retrieval task. To answer **RQ5.3**, we consider the similarity scores estimated for pairs of questions by different methods as features, use a learning to rank method to combine them, and compare its performance to each individual method.

6.4.1 Datasets

To train and evaluate the models proposed in Section 6.3, we use a Stack Overflow dataset (see footnote 1). This dataset contains programming related questions and answers and metadata associated with them. Information about duplicate/near-duplicate questions identified by users is also included in this dataset. To the best of our knowledge, this dataset is the only large and publicly available dataset in which questions have multiple fields. Although other datasets also have tags and answers associated with questions, they are relatively small and creating a neural model that exploits multiple fields is hard using these datasets.

We use pairs of questions labeled as “duplicate” and “near-duplicate” as positive samples. Users usually specify very relevant but not duplicate questions as “possible duplicates” in the comments. We extract “possible duplicate” questions from the dataset and consider them to be positive samples as well. As negative samples we select 500K questions and randomly select 1.5 million pairs of them. Table 6.1 contains the statistics of the dataset. We use title, body, and tags associated with questions together with their best answer for training CSA. In Stack Overflow terminology, the best answer for a question corresponds to the accepted answer by the questioner for the question. On the selected subset of Stack Overflow for training, the mean, median, and standard deviation of the length of the title of questions are 8, 8, and 4 words, respectively. The

Table 6.1: Statistics of the subset of the Stack Overflow dataset used for training models.

Label	Number of samples
Duplicate	515,051
Possible duplicate	376,201
Non-relevant	1,500,000

mean, median, and standard deviation of the length of the body of questions are 120, 92, and 112 words, respectively. For the length of answers, the mean, median, and standard deviation are 83, 58, and 91 words, respectively.

For evaluation, we select 1,000 questions that have at least three duplicate questions. For each question Q in this set, we consider its duplicates to be relevant questions. Moreover, as non-relevant questions, for each question Q, we randomly pick 20 questions from the selected subset of the Stack Overflow dataset: 10 questions that share at least one tag with Q and 10 questions without a common tag with Q. We evaluate different question retrieval methods on this dataset.

6.4.2 Pre-processing

We remove stopwords included in the standard stop word list from Python's NLTK package. We remove all HTML tags from textual fields. We consider each line of code as a separate sentence and do standard tokenization to extract tokens from them. For splitting sentences, we use NLTK's sentence tokenizer. We remove the 100 most frequent words in the collection and words with fewer than 5 occurrences. We do standard tokenization and apply the Porter stemmer to questions, answers, and tags.

6.4.3 Model parameters

We train a word2vec model on the Stack Overflow dataset. We use questions, their best answers, and tags for training the word2vec model. We apply the Skipgram architecture and remove words with fewer than 20 occurrences. We train an embedding with dimension 300 with a window size of 10. We use the trained embedding for creating word representations used in CSA. We set the number of hidden layers of the LSTM (forward and backward) model to 128. We set the initial learning rate to 10^{-3}. The batch size is set to 64. Training consists of 10 epochs. The size of the question embeddings (the first fully connected layer) is 64.

6.4.4 Baselines

We compare CSA to five baselines. We chose a question retrieval method that achieves state-of-the-art results [46] in this task; this method also addresses both the vocabulary gap and expressive inefficiency. We use some conventional lexical matching methods, some semantic matching models, and a learning to rank model that uses all fields of questions and the structure for matching as baselines.

Lexical matching. This baseline is based on the similarity of language models of questions. We construct language models of archived questions using their title, body, best answer, and tags. For query questions, we build language models using their title, body, and tags. The language models are estimated using maximum likelihood estimation and Jelinek-Mercer smoothing. We compute the similarity of language models of questions using KL-divergence.

Word2vec. We use the trained word2vec model and average embeddings of question words (body and title) and use it as their representation. Each question is represented as the weighted average of the embeddings of the words occurring in its title and body. Cosine is the used similarity measure.

Learning to rank. We extract a set of features for pairs of questions in the training set and build a Learning to Rank (LTR) model using them. We use LambdaMart to create a ranking model. To create training data we randomly select 1,000 questions that have at least two duplicate questions. For a question q in this set, we consider the duplicates of q to be relevant questions. Moreover, as non-relevant questions, for each query question q, we select 10 questions randomly from non-duplicate questions that share at least one tag with q and 10 questions that share no tag with q. We take different parts of query questions (title, body, and tags) and compute their similarity to different parts of archived questions (title, body, tags, and best answer). We compute the similarity between all fields in the query question and all fields in the archived question in three ways: language models with Jelinek-Mercer and Dirichlet smoothing and Okapi BM25. This yields 36 features (3 (for query questions) $\times$ 4 (for archived questions) $\times$ 3 (similarity measures)). The Jaccard similarity between tags is added as the 37-th feature.

Siamese Convolutional Neural Network. We use the Siamese Convolutional Neural Network for cQA (SCQA) model proposed by Das et al. [46] as a baseline. The results reported in the paper are state-of-the-art results on the Yahoo! Answers dataset. This method is a Siamese network that uses convolution layers to embed questions into a low-dimensional space in which cosine similarity of questions can be calculated. We use the hyper-parameters stated in [46]. The architecture contains three convolution layers followed by max-pooling and ReLU units. We use the dataset explained in §5.4.1 for training the SCQA model. Models of questions are based on all fields except the best answer. This field is excluded because it is very long for most of the questions and SCQA cannot encode it very accurately.

LSTM-based method. We use a bidirectional LSTM model to learn representations of questions.[4] Only the title field is used for training as it achieves better performance than using all fields. Each question in a pair of questions is represented as the concatenation of the forward and backward representations learned by the LSTM model. We concatenate representations of pairs of questions and feed them to a fully-connected layer to predict a relevance label. As the objective function, we use the cross-entropy loss. The number of hidden layers of the forward and backward LSTMs is set to 128. We initialize

[4] LSTMs have been used successfully for question retrieval. This particular architecture has been taken from https://engineering.quora.com/Semantic-Question-Matching-with-Deep-Learning. A similar approach is used in [202] for matching sentences.

the embeddings of words with a pre-trained word2vec model and update them during training.

6.4.5 Evaluation metrics

Question retrieval is a ranking task and ranking-based evaluation metrics have previously been used for this task [46, 142, 225, 231]. Mean Average Precision (MAP) is a common ranking metric for evaluating questions retrieval methods. Mean Reciprocal Rank (MRR) is another used metric that captures how far the first relevant question is in the ranking and how much effort should be put to find a relevant question in the returned ranking. R-Precision is used to capture recall and reflects how far should one go down in the ranked list to find all relevant questions. P@1 captures the quality of the top ranked question.

6.4.6 Statistical significance

For testing statistical significance of observed differences, we use a paired two-tailed t-test. We set α (the desired significance level) to 0.05; $^\blacktriangle$ and $^\blacktriangledown$ indicate that a method performs significantly better or worse than the corresponding baseline, respectively.

6.5 Results

Following the four research questions described in Section 6.4, we report the results of our proposed question retrieval method, CSA.

6.5.1 Question retrieval results

To answer **RQ5**, we use the evaluation setup described in Section 6.4.1 and report the results of different question retrieval methods on this dataset. Table 6.2 shows the results of this experiment. First, methods that use data fields in an intelligent way (CSA and LTR) are best performing. Although all other methods also use different data fields, they do not treat different parts of questions differently. Therefore, they are not able to learn the individual importance of different question fields.

Second, CSA significantly outperforms all baselines. Thus, automatically learning the weights of different parts of questions per sample by the attention mechanism is more effective than learning a global ranking function as done in the LTR method.

Third, for SCQA, we could not achieve the performance reported in [46]. We report the results of SCQA trained using all fields except the best answer field; including the best answer only further decreased performance. Unfortunately, the implementation details provided by the authors did not allow us to fully reproduce the authors' method.

Fourth, all methods have relatively good performance. This is due to the fact that we randomly selected non-relevant samples and in most cases there is a low similarity between the non-relevant samples compared to the duplicate question to the query questions. Therefore, distinguishing relevant from non-relevant questions is relatively easy.

Table 6.2: Results of different question retrieval methods on the Stack Overflow dataset. The significance test is done on the improvements of CSA over the closest competitor, that is, the learning to rank method.

Method	**MAP**	**P@1**	**R-Precision**	**MRR**
Language Model	0.71	0.70	0.62	0.76
Word Embedding	0.74	0.71	0.64	0.78
Learning to Rank	0.84	0.81	0.73	0.87
SCQA	0.63	0.59	0.51	0.68
LSTM	0.79	0.76	0.65	0.84
CSA	**0.88**$^{\blacktriangle}$	**0.85**$^{\blacktriangle}$	**0.78**$^{\blacktriangle}$	**0.91**$^{\blacktriangle}$

Table 6.3: Results of using different fields of questions as input. Ti, Bo, Ta, BA, and all stand for Title, Body, Tags, Best Answer, and all fields of questions, respectively.

Method	**MAP**	**P@1**	**R-Precision**	**MRR**
$Q_q(Ti), Q_c(Ti)$	0.73	0.68	0.61	0.75
$Q_q(Bo), Q_c(Bo)$	0.71	0.65	0.57	0.71
$Q_q(Ti), Q_c(Ti, Bo)$	0.77	0.73	0.65	0.80
$Q_q(Bo), Q_c(Ti, Bo)$	0.75	0.70	0.62	0.79
$Q_q(Ti), Q_c(Ti, Bo, BA)$	0.78	0.74	0.70	0.81
$Q_q(Bo), Q_c(Ti, Bo, BA)$	0.77	0.73	0.64	0.80
$Q_q(Ti), Q_c(all)$	0.81	0.78	0.72	0.83
$Q_q(Bo), Q_c(all)$	0.80	0.77	0.67	0.84
$Q_q(all), Q_c(Ti)$	0.81	0.77	0.70	0.84
$Q_q(all), Q_c(Bo)$	0.79	0.75	0.69	0.82
$Q_q(all), Q_c(Ti, Bo)$	0.83	0.80	0.73	0.87
$Q_q(all), Q_c(Ti, Bo, TA)$	0.86	0.83	0.75	0.90
$Q_q(all), Q_c(Ti, Bo, BA)$	0.85	0.81	0.73	0.88
$Q_q(all), Q_c(all)$	**0.88**	**0.85**	**0.78**	**0.91**

Fifth, the LSTM baseline, which only uses titles, performs better than the language model, word2vec, and SCQA methods. Adding more information to the input of the LSTM leads to a lower performance. Thus, titles are good indicators of the relevance of questions. As the results for CSA show, other fields are also informative but the LSTM is not able to exploit all of them in an effective way.

To analyze the effect of using different fields of questions on the performance of our method, we design another experiment in which we feed different combinations of fields as input to the model. Table 6.3 shows the results of this experiment. The results indicate that each field contributes to the achieved performance in the question retrieval task. The best performance is achieved when we use all fields of the query question and candidate questions. Titles of questions are in general more informative than other fields. We achieve a higher performance when we use titles of question than with other fields.

To sum up, using more data fields and rich content available for questions leads to a higher performance in the question retrieval task. CSA can effectively combine and

Table 6.4: Results of using different fields of questions as input to the CSA architecture without attention mechanism. $^{\blacktriangle}$ and $^{\blacktriangledown}$ indicate that the corresponding method performs significantly better or worse than its attention-enabled counterpart in Table 6.3.

Method	MAP	P@1	R-Precision	MRR
$Q_q(Ti), Q_c(Ti)$	$0.82^{\blacktriangle}$	$0.78^{\blacktriangle}$	$0.71^{\blacktriangle}$	$0.84^{\blacktriangle}$
$Q_q(Bo), Q_c(Bo)$	$0.79^{\blacktriangle}$	$0.74^{\blacktriangle}$	$0.66^{\blacktriangle}$	$0.81^{\blacktriangle}$
$Q_q(Ti), Q_c(Ti, Bo)$	$0.80^{\blacktriangle}$	$0.76^{\blacktriangle}$	$0.68^{\blacktriangle}$	$0.83^{\blacktriangle}$
$Q_q(Bo), Q_c(Ti, Bo)$	0.77	0.73	0.65	0.80
$Q_q(Ti), Q_c(Ti, Bo, BA)$	0.78	0.74	0.67	0.82
$Q_q(Bo), Q_c(Ti, Bo, BA)$	0.75	0.70	0.62	0.78
$Q_q(Ti), Q_c(all)$	$0.77^{\blacktriangledown}$	$0.72^{\blacktriangledown}$	$0.65^{\blacktriangledown}$	$0.80^{\blacktriangledown}$
$Q_q(Bo), Q_c(all)$	$0.76^{\blacktriangledown}$	$0.71^{\blacktriangledown}$	$0.62^{\blacktriangledown}$	$0.79^{\blacktriangledown}$
$Q_q(all), Q_c(Ti)$	$0.77^{\blacktriangledown}$	$0.73^{\blacktriangledown}$	$0.65^{\blacktriangledown}$	$0.81^{\blacktriangledown}$
$Q_q(all), Q_c(Bo)$	$0.76^{\blacktriangledown}$	$0.72^{\blacktriangledown}$	$0.64^{\blacktriangledown}$	$0.81^{\blacktriangledown}$
$Q_q(all), Q_c(Ti, Bo)$	$0.75^{\blacktriangledown}$	$0.72^{\blacktriangledown}$	$0.62^{\blacktriangledown}$	$0.81^{\blacktriangledown}$
$Q_q(all), Q_c(Ti, Bo, TA)$	$0.74^{\blacktriangledown}$	$0.70^{\blacktriangledown}$	$0.60^{\blacktriangledown}$	$0.78^{\blacktriangledown}$
$Q_q(all), Q_c(Ti, Bo, BA)$	$0.73^{\blacktriangledown}$	$0.70^{\blacktriangledown}$	$0.59^{\blacktriangledown}$	$0.77^{\blacktriangledown}$
$Q_q(all), Q_c(all)$	$0.73^{\blacktriangledown}$	$0.69^{\blacktriangledown}$	$0.58^{\blacktriangledown}$	$0.77^{\blacktriangledown}$

exploit the available information.

6.5.2 Attention mechanism

To answer **RQ5.1**, we design an experiment in which we train a variation of our model without attention mechanism. In this architecture, the input representation and output modules remain the same. However, in the multi-context attention module, the attention and linear functions are removed and all α_i's are set to 1. This removes the attention mechanism completely. The input to the aggregation function of candidate questions consists of representations of different fields without any attention to the context vectors.

Table 6.4 presents the results of this experiment. The performance of CSA without the attention mechanism (the last row in Table 6.4) drops dramatically and significantly. This demonsrtates the importance of the attention mechanism. Interestingly, the performance improves as we remove fields from questions. The best performance of CSA without attention mechanism is achieved when we only use titles of questions. This result is even better than the performance of the LSTM baseline, which indicates the ability of the proposed architecture to bridge the vocabulary gap. When we add more fields the performance drops. Adding more fields may end up adding noise to the input. Titles are usually accurate and very short, so even a simple architecture without attention mechanism can capture semantic similarity of questions through their titles. However, as we add more fields, the architecture cannot exploit extra information and address the expressive inefficiency challenge.

The performance of CSA (with attention mechanism) is significantly better than the performance of CSA without attention that only uses titles of questions. This result confirms that although titles are the most important parts of questions, other fields also contain useful information and adding them to an attention enabled model leads to

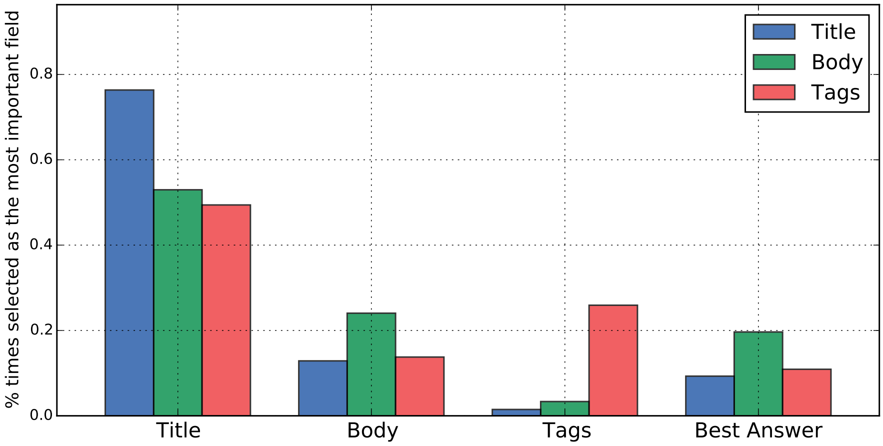


Figure 6.4: The relative number of times different fields of candidate questions in the evaluation set are selected as the most important field by the attention mechanism using different context vectors from query questions. The X-axis corresponds to the fields of candidate questions.

higher performance. However, without the attention mechanism it is not possible to add and exploit them in an effective way.

The results in this section confirm the importance of the attention mechanism and its ability to address expressive inefficiency by taking into account all valuable information associated with questions.

6.5.3 Analysis of the attention mechanism

To answer **RQ5.2**, we perform a quantitative analysis of the weights assigned by the attention mechanism to different fields of questions. Figure 6.4 shows the relative number of times different fields of candidate questions are selected as the most important field based on different context vectors obtained from query questions. This experiment is done on the evaluation set described in Section 6.4.1. The results show that the attention mechanism pays more attention to titles of candidate questions in general. This indicates that titles are the most informative parts of questions. A question's body and best answer are the next most important fields. Interestingly, when tags are used as context vectors, the attention mechanism pays more attention to tags of candidate questions than to their body and best answers. This result suggests that when using tags the attention mechanism tries to capture how relevant questions are topically and incorporate it in the matching mechanism.

6.5.4 Combining question retrieval methods

Next, we answer **RQ5.3** by combining different question retrieval methods. We take the scores generated by all baselines (lexical matching, word2vec, learning to rank, SCQA, and LSTM) and CSA and combine them in a learning to rank framework. We do 5-fold cross validation on the evaluation set. In each fold, we use 800 queries and

Table 6.5: Results of combining different question retrieval methods.

Method	**MAP**	**P@1**	**R-Precision**	**MRR**
CSA	0.88	0.85	0.78	0.91
Combination	0.92▲	0.89▲	0.83▲	0.94▲

Table 6.6: The performance drop (absolute difference) by removing different methods from the combination model.

Method	**ΔMAP**
CSA	−0.06
LTR	−0.03
LSTM	−0.01
word2vec	−0.01
Language model	−0.006
SCQA	−0.002

their associated candidate questions for training the learning to rank model and 200 queries of evaluation. We use LambdaMart for creating the ranking model.

Table 6.5 shows the results of this experiment, next to the results of CSA. The results show that adding other methods on top of CSA leads to a significant improvement. This indicates that CSA and the other methods are capturing different signals about question relevance. To gain more insights into the contribution of each individual method on the results achieved by the learned combination model, we perform an additional analysis. We remove methods from the combination model one by one and measure the drop in performance in terms of MAP; see Table 6.6. When CSA is removed from the model the performance drops most. Moreover, removing the LTR model also causes a considerable drop in performance. Removing other features does not have a significant effect on the final performance. This shows that CSA and LTR are the most important features in the combination and are capturing different information about the relevance of questions.

6.6 Related Work

We review related work from four angles: question retrieval, short text embeddings, attention mechanisms, and evaluation.

6.6.1 Question retrieval

Relevant question retrieval is a task associated with cQA sites [142, 152, 178]. Traditional approaches for question retrieval are term matching based methods such as BM25 and language models [92, 96, 139, 178]. Term matching methods have a poor performance on the question retrieval task [152, 178]. The main reason is the vocabulary gap [46, 178, 231]. To overcome this problem, many approaches have been proposed that try to capture the semantics of questions and match questions based on that. An early example are translation models, which treat question/answer pairs as parallel sentences and extract semantically similar word/phrase pairs. Translation-based question matching models outperform traditional term matching models [34, 211, 217, 229]. Syntactic tree-based methods are another type of approach that try to augment term matching and translation-based models with syntactic information included in questions [198–200]. Other semantic resources such as entities contained in questions and user intent information have also been exploited to augment term matching approaches [174, 230].

Recently, neural network-based approaches have been adapted to learn semantic representations of questions and use those for question retrieval [27, 46, 149, 157, 201, 231, 232]. Das et al. [46] propose a Siamese network that consists of twin convolutional neural networks (CNNs) with shared parameters for learning question representations. Qiu and Huang [157] use a similar approach for matching questions and answers for finding most relevant answers of questions. Zhou et al. [231] propose a model that can learn word embeddings empowered by category information within cQA sites. Similar to the skip-gram model [138], they use co-occurrence information of words to learn embeddings. Wang et al. [201] propose a similar approach for learning question representations; they also empower the skip-gram model with category information and use a CNN to learn representations of questions based on word embeddings.

Most neural approaches only use titles of questions for estimating the semantic similarity. Only a few have tried to use more information such as tags associated with questions [201, 231]. To the best of our knowledge, we are the first to exploit multiple data fields available for questions such as a question's title/body, answers, and tags in a unified way for learning the representations of questions.

We perform our experiments on the Stack Overflow dataset which contains programming related questions which are a mixture of text and code. In this regard, our task is also related to source code retrieval task in which given a piece of code the goal is to find similar code snippets from a collection [140, 153]. Our task however differs as we deal with a mixture of code and text, while in the code retrieval task, the query code is a piece of code.

6.6.2 Short text embeddings

In this study, we learn embeddings of questions based on the embedding of their words. There are different approaches for aggregating word embeddings into sentence/document embeddings. The simplest approach is taking the average of word embeddings [2, 222]. This approach cannot capture all information included in documents such as frequency of words and word order. More advanced methods have been proposed to learn to combine embeddings of words. E.g., De Boom et al. [48] incorporate the frequency of words in learning the composition; Kenter et al. [105] use the average operator on word embeddings to estimate sentence embeddings. CNNs and recurrent neural networks (RNNs) are also used for aggregating word embeddings [38, 101]. The Fisher Vectors method is a popular approach for aggregation too [40, 231]. This approach assumes that word embeddings are generated by a universal probability density function that is assumed to be a Gaussian mixture model.

We take a hierarchical approach to embedding short texts. To avoid complexity, we represent sentences by the average of embeddings of their words. Unlike previous work, at the document level, we learn a representation based on a recurrent representation of sentences.

6.6.3 Neural attention mechanisms

In this work, we use an attention mechanism for weighting different parts of questions. Attention mechanisms are mostly used in the context of machine translation in which there is an encoder and a decoder [204]. Attention mechanisms have previously been used for question answering and reading comprehension [78, 181]. Here, the goal is to read a set of sentences and then answer a question relevant to those sentences. Attention helps to find the most relevant sentences to the question and generate the answer. Similar to this work, we use an attention mechanism for matching questions. However, our setting is different from the reading comprehension task in important ways. We use multiple data fields to estimate similarity and use an attention mechanism as a weighting function to choose relevant parts of questions rather than having one data field and selecting answers of questions using an attention mechanism. Therefore, the goal, input, and outputs of the two tasks differ, and, hence, so do the attention mechanisms. To the best of our knowledge, there is no existing work on using an attention mechanism for capturing text similarity using multiple data fields.

In general, attention mechanisms have a great potential to automatically ignore the unimportant parts in the entire input sequence and highlighting important parts. In the case of cQA, Hsu et al. [85] use an attention mechanism for computing the similarity of questions and answers. A similar approach has been used in [184], where an attention mechanism is used for matching questions and answers.

We use a similar approach to Hsu et al. [85]. An important difference, however, is that we use multiple context vectors to weight different parts of questions and combine the achieved attention signals in a principled way. Our attention mechanism is very different from the conventional attention mechanism used for machine translation and relevant tasks. Instead of using one context vector, it uses multiple context vectors and in the end weights and combines the representation obtained from multiple fields based on different context vectors.

6.6.4 Evaluating question retrieval methods

Most question retrieval methods are trained and evaluated on Yahoo! Answers and Stack Exchange datasets [142, 178]. Among existing datasets, only these two datasets have multiple data fields associated with them. However, no ground-truth is available for the Yahoo! Answers dataset. Zhang et al. [224] made a test collection on this dataset by annotating pairs of questions and assigning ground-truth labels to them. This evaluation set has been used in several studies [46, 224, 225]. The created dataset is relatively small, making it very hard to train a generalizable state-of-the-art neural model. Unlike the Yahoo! Answers dataset, the Stack Exchange dataset contains information of duplicate/near-duplicate questions identified by users, which can be directly used for training and evaluating question retrieval models. Recently, this dataset has been used in the SemEval-2017 competition [142]. Therefore, in this study, we use the Stack Exchange dataset for creating our model.

We are the first to build an evaluation set using Stack Overflow dataset that contains questions with multiple data fields.

6.7 Conclusion

We have proposed CSA (Question Retrieval using Content, Structure and Attention), a method for finding relevant questions on cQA forums based on different data fields associated with questions. CSA is built around a multi-context attention mechanism to combine all information associated with questions and learn their representations. We have shown that CSA is able to exploit rich content and structure available on cQA forums and address two main challenges with the question retrieval task: the *vocabulary gap* and *expressive inefficiency*.

Our main findings are the following: (1) Semantic matching models are quite successful in bridging the vocabulary gap problem for the question retrieval task. These methods achieve a good performance using only titles of questions. Using different data fields beside the title, we can further bridge the vocabulary gap and improve the performance of title-only based approaches. (2) Expressive inefficiency is one of the big challenges in finding relevant questions. Using rich content and structure of questions can help to address this problem. None of the existing semantic matching models can effectively exploit multiple data fields available for questions. A multi-context attention mechanism can overcome this problem by adjusting the weights of different fields of questions while learning their representation. (3) When we combine lexical matching with semantic matching models for question retrieval, we achieve even higher performances. This result confirms that semantic and lexical matching models provide complementary signals.

The main limitation of our work is the way our evaluation set is constructed. Since there are no available datasets for evaluating question retrieval methods that contain multiple data fields for questions, we had to construct an evaluation set automatically using information about duplicate questions. A more natural way would to construct an evaluation set using a pooling approach.

As to future work, for input representation, we represented textual fields as sequences of sentences in which each sentence representation is the average of embeddings of their words; a more principled alternative would be to have an attention mechanism to further weight sentences and words based on the amount of information they contain and use the weighted combination for learning question representation. Moreover, in CSA we projected all data fields into the same embedding space. A possible future direction would be to learn a separate representation for each field and then learning a transformation between these spaces. This would let the model capture characteristics of each field even more accurately.

This chapter was the last chapter in this thesis. We studied how metadata and structure can be useful in exploratory search. Next, we will conclude the thesis and elaborate on possible future work in this direction.

7 Conclusions

In this chapter, we first revisit our research questions introduced in Chapter 1 and summarize main findings and implications of our research in Section 7.1. Then, in Section 7.2, we describe the main limitations of our work and the possible future directions.

7.1 Main Findings

7.1.1 Multi-label text classification

We started with the task of Multi-label Text Classification (MLTC) and asked:

RQ1 How helpful is integrating a variety of sources of information extracted from content, structure, and metadata as features to improve the performance of MLTC systems?

To answer this question, we considered MLTC, a ranking problem in which the goal was ranking class labels given a document. We used a Learning to Rank (LTR) approach to rank classes and estimate a classifier: we simply rank all classes given a document and assign the top k classes (k to be determined by another classifier) as labels to the input document. To exploit structure and metadata, we used different sources of information associated with documents and classes to extract features which reflect the similarity of documents and classes. In addition to the associated explicit information with classes, we used co-occurrence patterns of classes as labels of documents as another source of information. This is crucial information for MLTC as the main characteristic of this task is that documents have multiple class labels and there are implicit relations between the classes.

Starting from simple classification approaches for MLTC such as support vector machines, we observed that these approaches are not optimal for the task of MLTC when there is a high number of classes. They cannot model the underlying relations between classes and they are very inefficient. We showed by experimental means that without even using all information sources, ranking-based approaches achieve a reasonable performance for this task and outperform simple classifiers. Our proposed LTR approach further improved the performance of ranking-based models by integrating all features extracted from all sources of information. We performed a feature analysis to determine the effectiveness of different source of information for MLTC. We observed that, labeled training documents is the primary and most effective source of information. However, exploiting structure of documents, we interestingly, found that it is more effective to use the structure and treat different parts of documents differently rather than considering all parts of documents a single text. Moreover, we found that other sources of information such as the hierarchical thesaurus of classes contribute to the performance of the designed classifier. We showed that we can use the outcomes of this feature analysis to design a lean-and-mean classifier which is both efficient and effective. Then, we focused on using implicit co-occurrence patterns of classes and found that using this information also helps in classifying documents more accurately. Interestingly, we found that the co-occurrence information is more helpful when the underlying classifier has a poor performance. We studied different aspects of the MLTC task, including the effect of size of the training set and a dynamic method for selecting the number of classes. We found that ranking-based models are more robust against the size of the dataset compared to the classification-based approaches. Our analysis on the use of a dynamic approach for selecting the number of classes revealed that the better the underlying classifier, the more the dynamic model can further improve performance. These findings indicate that ranking-based models are better alternatives in the MLTC task.

7.1.2 Measuring topical diversity of documents

Similar to previous work, we used topic models for measuring topical diversity. We characterized two drawbacks with using topic models for measuring topical diversity and proposed a method to address them and answered our second research question:

RQ2 How effective is our hierarchical re-estimation approach in measuring topical diversity of documents? Are the topic models that have been improved in this way also successfully applicable for other tasks such as documents clustering and classification?

We proposed a hierarchical re-estimation method aimed at removing the two issues associated with topic models which make them less suitable for measuring diversity: generality and impurity. The quality of topic models depends on three distributions: words in documents, words in topics and topics in documents. We showed that distributions of words in documents and topics in documents suffer from the generality problem. On the other hand, both distributions of words in topics and topics in documents suffer from the impurity issue. Based on this finding, we proposed three re-estimation approaches aimed at removing generality and impurity from the mentioned distributions. Our analysis indicated that generality is the more important issue with topic models when they are used for the task of measuring topical diversity. This shows that distributions of words in documents and topics in documents are more important in the topical diversity task and to achieve a good performance they should be modeled accurately. Although impurity has less negative impact on the performance, removing it helps to achieve more improvements. We found that although our re-estimation approach was originally proposed for the topical diversity task, it has a good performance in other document management tasks such as classification and clustering. Therefore, the improved topic models achieved using the re-estimation approach can be seen as a generic framework for improving the performance in any document management task that uses topic models.

7.1.3 Commitment detection in email

Our third research question was:

RQ3 Can commitments be reliably detected in emails? And how does the performance of commitment models change when they are tested on a different domain than they are trained on?

Our findings indicate that there is a domain bias associated with email datasets and it can affect the performance and generalizability of any model trained on them. We cannot train a model on one dataset and apply it reliably on a different domain.

We created a large-scale dataset with commitment labels from two publicly available email datasets. We characterized the core language of commitments using a feature selection approach. We then modeled commitment detection as a binary classification task (on sentences) and used logistic regression to check how well one can detect commitments. We found that, indeed, a simple classifier can detect commitments effectively when we train and test them on the same domain (email dataset). We argued that the publicly available datasets are skewed in different ways: they have very specific focus areas and they are quite old. To see if this has a negative impact on the performance of the trained commitment model, we evaluated its performance across domains and found that domain difference has a strong negative impact on performance. Interestingly, we found that we can characterize the domain difference using some simple feature analysis. We used this characterization to remove domain information from email datasets and design domain-independent commitment models. We compared the effectiveness of domain adaptation methods working on different levels (feature-level and sample-level) and found that domain adaptation can undo the domain bias present in email datasets. We designed a deep autoencoder to do feature-level and sample-level adaptation jointly and showed that it can outperform existing adaptation models in commitment detection task.

7.1.4 Computing semantic shifts

In the second part of the thesis, we focused on using metadata and structure for improving the performance in two exploratory search tasks. The first task is detecting semantic shifts across viewpoints. We were interested in using metadata features such as time and groups associated with documents for detecting shifts in meaning of words. Our fourth research questions was:

RQ4 How can we use metadata information to measure semantic shifts? How effective are the proposed approaches in quantifying the changes in word meaning over various dimensions such as time and political context?

We argued that semantic shifts can occur both over a long period of time and in a short period of time across groups of people with diverging viewpoints. The existence of shifts in meanings over a long period of time is known from the literature [118, 175, 187, 210]. We focused on uncovering shifts in a short period of time. We performed an analysis on the UK parliamentary proceedings and found that indeed shifts can occur in a short period of time across groups which have diverging viewpoints. Then, we focused on measuring such shifts. We used distributional semantics to extract meaning of words and compared several methods to use these meanings to detect semantic shifts. We showed that the proposed methods are very successful in discovering shifts on political speeches and news. Moreover, we used information about semantic shifts to boost performance in ideology detection and viewpoint summarization. We found that temporal laws of change also apply to other dimensions. This indicates that the meaning of frequent words do not shift across viewpoints, while even the meaning of words with low polysemy values can shift. Furthermore, we proposed another law for semantic shifts which implies that more concrete words are insensitive to the viewpoint of speaker.

7.1.5 Question retrieval in cQA forums

Finally, we focused on finding similar questions in cQA forums and boosting performance using content, metadata, and structure. Our last research question was:

RQ5 Does using the rich data and structure available on question answering forums lead to a better score than the state of the art on the task of question retrieval?

We argued that two main issues associated with question retrieval are the vocabulary gap and expressive inefficiency problems. Existing semantic matching methods are successful in bridging the vocabulary gap to some extent. Most of them are only using the title of questions for matching since other fields are quite long and these methods do not have an effective mechanism to handle large inputs and multiple fields. Based on our analysis, we found that expressive inefficiency is still a hard problem to handle for existing question retrieval methods. This is another reason why existing models only use titles since titles are usually more accurate that other fields. Inspired by recent advancements in the machine reading comprehension task [78, 181], we used an attention mechanism to control the contribution of different fields of questions for computing their semantic similarity. We showed that this approach can further bridge the vocabulary gap since it is using more information beside titles. Moreover, by controlling the contribution of fields, the attention mechanism automatically controls the noise and addresses the expressive inefficiency problem. We further combined similarity scores estimated for questions using different question retrieval methods in a learning to rank framework and found that when we combine lexical matching with semantic matching models, we achieve even higher performances. This result confirms that semantic and lexical matching models provide complementary signals.

7.2 Future Work

We list possible directions for future work grouped by our main research questions.

7.2.1 Optimizing a ranking loss for MLTC

Based on our results simple classification approaches (such as binary classification methods) for MLTC fail when there is a high number of classes and ranking-based approaches outperform classification-based systems. Moreover, rather than just using a set of training samples, utilizing various sources of information leads to better performances in MLTC. This opens up an interesting direction to design more effective MLTC systems by defining and optimizing a ranking loss rather than a classification loss which can integrate multiple sources of information. Our work was based on extracting features from document-class pairs and learning a ranking function on top of them. It would be interesting to use raw document-class features to optimize a ranking loss. This can be done using a neural model which takes different fields of documents together with other sources of information as input, encodes them and produces scores per class. A ranking loss then can be used to train the model.

7.2.2 Removing specificity from topic models

Our topic models re-estimation approach is most effective in removing general information from probability distributions. Another issue with topic models is the existence of very specific information in them. Specific information is not representative of the collection and it can lower the purity of topic models. Therefore, to train a more accurate topic model which has a good performance in topical diversity task it is also important to remove very specific words from documents. Current approaches, including ours, are not able to address this problem adequately. An interesting direction would be designing re-estimation methods that can remove undesired specific information from probability distributions.

7.2.3 Contextualizing commitment models

We modeled commitment detection as a binary classification task: does a given sentence from an email contain a commitment or not? For simplicity and generalizeability reasons, we only used sentences as samples and did not use the context around them. The context such as the rest of the email and previous emails could potentially be useful to disambiguate the input sentence and detect commitments more accurately. In addition to this information, the information about the sender and receiver of emails, time of sending the email, and history of interaction of the sender and the receiver can be further used for detecting commitments. Future work could integrate all this information in a unified way to learn a commitment model from it. This can be done by learning a representation for each data field (e.g., sentence itself, whole email, sender, etc), computing a probability of constituting a commitment using each field, and weighting and integrating them to predict the class. In addition to this, it would be interesting to see if this additional information can help make a more robust and domain-independent commitment model.

7.2.4 Applications of stability measures

The estimated stability measures in Chapter 5 can be useful in various applications. We used them in two extrinsic tasks, e.g., summarizing diverging viewpoints and ideology detection defined in terms of document classification. The generated summaries using the stability measures can be used in exploratory search scenario to uncover diverging aspects of a given topic. We only focused on detecting shifts in political and media discourse, but our approaches are applicable in any other kind of discourse such as different groups in social media. The input would be two or more groups and for each group a collection of text and our measures can automatically detect topics for which the groups have different viewpoints and a summary of viewpoints of each group regarding the topic. We can also broaden the set of applications of stability measures, by, for example, examining how our approach contributes to other tasks such as controversy detection and locating people in a ”filter bubble”. If the language use of a specific group exhibits radically divergent word meanings, then they might be in a filter bubble and word stability can be used to quantify this.

7.2.5 Learning semantic spaces for different fields

For representing the input in the question retrieval task, we represented textual fields as sequences of sentences in which each sentence representation is the average of embeddings of their words. The embeddings are learned during training the model. In this approach, there is no weighting and attention mechanism for selecting most informative words in sentences. A more principled alternative would be to have an attention mechanism to further weight sentences and words based on the amount of information they contain and use the weighted combination for learning question representation. Moreover, in our approach, we projected all data fields into the same embedding space. However, the nature of the fields are somewhat different. For example, tags are very different than textual fields such as title, body, and answers. In addition to the used fields, if we want to use other fields such as time of creation, questioner and answerer information, it does not make sense to project all the fields to the same embedding space. To address this issue, a possible future direction would be to learn a separate representation for each field. Then, a transformation between these spaces can be learned to make it possible to compute similarities on different spaces. This would let the model capture characteristics of each field even more accurately.

Bibliography

[1] National Center for Biotechnology Information, U.S. National Library of Medicine. Pubmed Central Open Access Initiative. 2010. (Cited on page 43.)

[2] S. Arora, Y. Liang, and T. Ma. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of International Conference on Learning Representations*, ICLR '17, 2017. (Cited on page 120.)

[3] R. Artstein and M. Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008. (Cited on page 64.)

[4] H. Azarbonyad. Measuring interestingness of political documents. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 1175–1175, 2016. (Cited on page 9.)

[5] H. Azarbonyad and E. Kanoulas. Power analysis for interleaving experiments by means of offline evaluation. In *Proceedings of the 2017 ACM International Conference on the Theory of Information Retrieval*, ICTIR '17, pages 87–90, 2016. (Cited on page 9.)

[6] H. Azarbonyad, M. Dehghani, M. Marx, and J. Kamps. Time-aware authorship attribution for short text streams. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 727–730, 2015. (Cited on page 9.)

[7] H. Azarbonyad, F. Saan, M. Dehghani, M. Marx, and J. Kamps. Are topically diverse documents also interesting? In *Proceedings of the 6th International Conference on Experimental IR Meets Multilinguality, Multimodality, and Interaction*, CLEF '15, pages 215–221, 2015. (Cited on pages 7, 33, 36, and 42.)

[8] H. Azarbonyad, M. Dehghani, K. Beelen, A. Arkut, M. Marx, and J. Kamps. Words are malleable: Computing semantic shifts in political and media discourse. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '17, pages 1509–1518, 2017. (Cited on pages 8, 66, and 81.)

[9] H. Azarbonyad, M. Dehghani, T. Kenter, M. Marx, J. Kamps, and M. de Rijke. Hierarchical re-estimation of topic models for measuring topical diversity. In *Proceedings of the 39th European Conference on IR Research*, ECIR '17, pages 68–81, 2017. (Cited on pages 7 and 17.)

[10] H. Azarbonyad, M. Dehghani, T. Kenter, M. Marx, J. Kamps, and M. de Rijke. HiTR: Hierarchical topic model re-estimation for measuring topical diversity of documents. *IEEE Transactions on Knowledge and Data Engineering, to appear*, 2018. (Cited on pages 7 and 33.)

[11] H. Azarbonyad, M. Dehghani, M. Marx, and M. de Rijke. Learning question representations for question retrieval using content, structure and attention. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, under review, 2018. (Cited on pages 8 and 103.)

[12] H. Azarbonyad, M. Dehghani, M. Marx, and J. Kamps. Learning to rank for multi label text classification: Combining different sources of information. *Journal of Natural Language Engineering, under review*, 2018. (Cited on pages 7 and 13.)

[13] H. Azarbonyad, R. Sim, and R. W.White. Domain adaptation for commitment detection in email. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, 2018. (Cited on pages 7 and 59.)

[14] R. Babbar and B. Schölkopf. Dismec: Distributed sparse machines for extreme multi-label classification. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 721–729, 2017. (Cited on page 15.)

[15] K. Bache, D. Newman, and P. Smyth. Text-based measures of document diversity. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, SIGKDD '13, pages 23–31, 2013. (Cited on pages 1, 3, 33, 36, 41, 43, and 44.)

[16] O. Becher, L. Hollink, and D. Elliott. Exploring concept representations for concept drift detection. In *SEMANTiCS 2017 Workshop Proceedings: Drift-a-LOD*, 2017. (Cited on page 84.)

[17] V. Bellotti, N. Ducheneaut, M. Howard, and I. Smith. Taking email to task: the design and evaluation of a task management centered email tool. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, SIGCHI '03, pages 345–352, 2003. (Cited on pages 2, 3, 59, and 61.)

[18] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010. (Cited on page 62.)

[19] Y. Bengio. Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, 2(1): 1–127, 2009. (Cited on pages 60 and 68.)

[20] P. N. Bennett and J. Carbonell. Detecting action-items in e-mail. In *Proceedings of the 28th Interna-*

tional ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05, pages 585–586, 2005. (Cited on pages 2, 3, 59, and 61.)

[21] W. Bi and J. T. Kwok. Multi-label classification on tree and dag-structured hierarchies. In *Proceedings of the 28th International Conference on Machine Learning*, ICML '11, pages 17–24, 2011. (Cited on page 16.)

[22] W. Bi and J. T. Kwok. Efficient multi-label classification with many labels. In *Proceedings of the 30th International Conference on Machine Learning*, ICML '13, pages 405–413, 2013. (Cited on pages 13 and 15.)

[23] D. M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, 2012. (Cited on page 39.)

[24] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. (Cited on pages 35, 36, and 39.)

[25] E. Bleich, H. Nisar, and R. Abdelhamid. The effect of terrorist events on media portrayals of islam and muslims: evidence from new york times headlines, 1985–2013. *Ethnic and Racial Studies*, 39(7): 1109–1127, 2016. (Cited on page 81.)

[26] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, ACL '07, pages 440–447, 2007. (Cited on page 60.)

[27] D. Bonadiman, A. Uva, and A. Moschitti. Multitask learning with deep neural networks for community question answering. *arXiv preprint arXiv:1702.03706*, 2017. (Cited on page 120.)

[28] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Proceedings of Advances in Neural Information Processing Systems*, NIPS '16, pages 343–351, 2016. (Cited on page 62.)

[29] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004. (Cited on page 16.)

[30] J. Boyd-Gaber, D. Mimno, and D. Newman. Care and feeding of topic models: Problems, diagnostics, and improvements. In *Handbook of Mixed Membership Models and Their Applications*. CRC Press, 2014. (Cited on pages 35 and 40.)

[31] M. Brysbaert, A. B. Warriner, and V. Kuperman. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911, 2014. (Cited on page 99.)

[32] L. Cai, G. Zhou, K. Liu, and J. Zhao. Learning the latent topics for question retrieval in community QA. In *Proceedings of 5th international joint conference on Natural Language Processing*, IJCNLP '11, pages 273–281, 2011. (Cited on page 104.)

[33] R. Campos, G. Dias, A. M. Jorge, and A. Jatowt. Survey of temporal information retrieval and related applications. *ACM Computing Surveys (CSUR)*, 47(2):15, 2015. (Cited on page 1.)

[34] X. Cao, G. Cong, B. Cui, C. S. Jensen, and Q. Yuan. Approaches to exploring category information for question retrieval in community question-answer archives. *ACM Transaction on Information Systems*, 30(2):7:1–7:38, 2012. (Cited on page 120.)

[35] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-graber, and D. M. Blei. Reading tea leaves: How humans interpret topic models. In *Proceedings of Advances in Neural Information Processing Systems*, NIPS '09, pages 288–296, 2009. (Cited on page 36.)

[36] M. Chen, Z. Xu, K. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*, 2012. (Cited on pages 62 and 68.)

[37] X. Chen and C. Cardie. Multinomial adversarial networks for multi-domain text classification. *arXiv preprint arXiv:1802.05694*, 2018. (Cited on page 62.)

[38] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. (Cited on pages 70, 110, and 120.)

[39] A. Clare and R. D. King. Knowledge discovery in multi-label phenotype data. In *Proceedings of the European Conference on Machine Learning & Principles and Practice of Knowledge Discovery*, ECML-PKDD '01, pages 42–53, 2001. (Cited on page 16.)

[40] S. Clinchant and F. Perronnin. Aggregating continuous word embeddings for information retrieval. In *Workshop on Continuous Vector Space Models and their Compositionality*, pages 100–109, 2013. (Cited on page 120.)

[41] W. W. Cohen, V. R. Carvalho, and T. M. Mitchell. Learning to classify email into"speech acts". In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '04, 2004. (Cited on pages 3, 59, 61, 62, and 72.)

[42] C. Constantinopoulos, M. Titsias, and A. Likas. Bayesian feature and model selection for gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):1013–1018,

2006. (Cited on page 36.)

[43] S. Corston-Oliver, E. Ringger, M. Gamon, and R. Campbell. Task-focused summarization of email. *Text Summarization Branches Out*, 2004. (Cited on pages 2, 3, 59, 61, 62, and 63.)

[44] M. Cutler, Y. Shih, and W. Meng. Using the structure of HTML documents to improve retrieval. In *USENIX Symposium on Internet Technologies and Systems*, pages 241–252, 1997. (Cited on page 1.)

[45] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 193–200, 2007. (Cited on pages 62 and 77.)

[46] A. Das, H. Yenala, M. Chinnakotla, and M. Shrivastava. Together we stand: Siamese networks for similar question retrieval. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, pages 378–387, 2016. (Cited on pages 4, 5, 103, 104, 113, 114, 115, 119, 120, and 121.)

[47] V. Daudaravicius. Automatic multilingual annotation of EU legislation with Eurovoc descriptors. In *Proceedings of Exploring and Exploiting Official Publications Workshop Programme*, EEOP2012, pages 14–20, 2012. (Cited on page 18.)

[48] C. De Boom, S. Van Canneyt, T. Demeester, and B. Dhoedt. Representation learning for very short texts using weighted word embedding aggregation. *Pattern Recognnition Letters*, 80(C):150–156, 2016. (Cited on page 120.)

[49] L. M. de Campos and A. E. Romero. Bayesian network models for hierarchical text classification from a thesaurus. *International Journal of Approximate Reasoning*, 50(7):932–944, 2009. (Cited on page 17.)

[50] R. De Felice. *A corpus-based classification of commitments in Business English*, pages 153–171. Springer Netherlands, 2013. (Cited on pages 3, 59, 61, and 72.)

[51] M. Dehghani, H. Azarbonyad, M. Marx, and J. Kamps. Sources of evidence for automatic indexing of political texts. In *Proceedings of the 37th European Conference on IR Research*, ECIR '15, pages 568–573, 2015. (Cited on pages 7 and 85.)

[52] M. Dehghani, H. Azarbonyad, J. Kamps, D. Hiemstra, and M. Marx. Luhn revisited: Significant words language models. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '16, pages 1301–1310, 2016. (Cited on page 9.)

[53] M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. Two-way parsimonious classification models for evolving hierarchies. In *Proceedings of the 7th International Conference on Experimental IR Meets Multilinguality, Multimodality, and Interaction*, CLEF '16, pages 69–82, 2016. (Cited on pages 9, 17, and 85.)

[54] M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. Generalized group profiling for content customization. In *Proceedings of the ACM SIGIR Conference on Human Information Interaction and Retrieval*, CHIIR '16, pages 245–248, 2016. (Cited on page 9.)

[55] M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. On horizontal and vertical separation in hierarchical text classification. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ICTIR '16, pages 185–194, 2016. (Cited on pages 9, 17, and 85.)

[56] M. Dehghani, H. Azarbonyad, J. Kamps, and M. Marx. Significant words language models for contextual suggestion. In *Proceedings National Institute for Standards and Technology*, 2016. (Cited on page 9.)

[57] M. Dehghani, H. Azarbonyad, J. Kamps, and M. de Rijke. Share your model instead of your data: Privacy preserving mimic learning for ranking. In *SIGIR wokshop on Neural Information Retrieval*, 2017. (Cited on page 8.)

[58] M. Dehghani, G. Jagfeld, H. Azarbonyad, A. Olieman, J. Kamps, and M. Marx. On search powered navigation. In *Proceedings of the 2017 ACM International Conference on the Theory of Information Retrieval*, ICTIR '17, pages 317–320, 2017. (Cited on page 8.)

[59] M. Dehghani, G. Jagfeld, H. Azarbonyad, A. Olieman, J. Kamps, and M. Marx. Telling how to narrow it down: Browsing path recommendation for exploratory search. In *Proceedings of the ACM SIGIR Conference on Human Information Interaction and Retrieval*, CHIIR '17, pages 369–372, 2017. (Cited on page 8.)

[60] M. Dehghani, H. Azarbonyad, J. Kamps, and M. de Rijke. Learning to transform, combine, and reason in open-domain question answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, 2018. (Cited on page 8.)

[61] M. Derzinski and K. Rohanimanesh. An information theoretic approach to quantifying text interestingness. In *NIPS MLNLP workshop*, 2014. (Cited on pages 1, 3, 33, and 36.)

[62] A. Elisseeff and J. Weston. A kernel method for multi-labelled classification. In *Proceedings of*

Advances in Neural Information Processing Systems, NIPS '01, 2001. (Cited on pages 16 and 26.)

[63] EuroVoc. Multilingual thesaurus of the european union, 2014. `http://eurovoc.europa.eu/`. (Cited on pages 21 and 23.)

[64] Y. Fang, L. Si, N. Somasundaram, and Z. Yu. Proceedings of the fifth acm international conference on web search and data mining. In *WSDM '12*, pages 63–72, 2012. (Cited on page 85.)

[65] A. Fauzan and M. L. Khodra. Automatic multilabel categorization using learning to rank framework for complaint text on bandung government. In *Proceedings of the International Conference of Advanced Informatics: Concept, Theory and Application*, ICAICTA '14, pages 28–33, 2014. (Cited on page 17.)

[66] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker. Multilabel classification via calibrated label ranking. *Machine learning*, 73(2):133–153, 2008. (Cited on page 16.)

[67] W. B. Gallie. Essentially contested concepts. In *Proceedings of the Aristotelian society*, volume 56, pages 167–198, 1955. (Cited on page 81.)

[68] M. Gamon, S. Azzam, Y. Cai, N. Caldwell, and Y.-Y. Wang. Automatic task extraction and calendar entry, 2013. US Patent App. 13/170,660. (Cited on page 77.)

[69] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014. (Cited on page 62.)

[70] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016. (Cited on pages 62 and 63.)

[71] L. M. Garshol. Metadata? Thesauri? Taxonomies? Topic maps! Making sense of it all. *Journal of information science*, 30(4):378–391, 2004. (Cited on page 1.)

[72] N. Ghamrawi and A. McCallum. Collective multi-label classification. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 195–200, 2005. (Cited on page 16.)

[73] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, ICML '11, pages 513–520, 2011. (Cited on page 62.)

[74] J. Gwizdka. Reinventing the inbox: Supporting the management of pending tasks in email. In *SIGCHI'02 Extended Abstracts on Human Factors in Computing Systems*, pages 550–551, 2002. (Cited on pages 59 and 61.)

[75] W. L. Hamilton, J. Leskovec, and D. Jurafsky. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, pages 1489–1501, 2016. (Cited on pages 4, 81, 82, 83, 84, 86, 91, and 99.)

[76] W. L. Hamilton, J. Leskovec, and D. Jurafsky. Cultural shift or linguistic drift? Comparing two computational measures of semantic change. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '16, pages 2116–2121, 2016. (Cited on pages 84 and 91.)

[77] B. Hariharan, L. Zelnik-manor, S. V. N. Vishwanathan, and M. Varma. Large scale max-margin multi-label classification with priors. In *Proceedings of the 27th International Conference on Machine Learning*, ICML '10, pages 423–430, 2010. (Cited on page 16.)

[78] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom. Teaching machines to read and comprehend. In *Proceedings of Advances in Neural Information Processing Systems*, NIPS '15, pages 1693–1701, 2015. (Cited on pages 121 and 126.)

[79] F. Herrera, F. Charte, A. J. Rivera, and M. J. del Jesus. *Multilabel classification: Problem analysis, metrics and techniques*, pages 17–31. Springer International Publishing, 2016. (Cited on pages 2 and 13.)

[80] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, 2004. (Cited on pages 34, 36, 37, and 44.)

[81] G. Hirst, Y. Riabinin, J. Graham, and M. Boizot-roche. Text to ideology or text to party status? In *Proceedings of Document Analysis in Political Science, T2PP Workshop*, pages 93–116, 2014. (Cited on page 85.)

[82] T. K. Ho, L. A. Lastras, and O. Shmueli. Concept evolution modeling using semantic vectors. In *Proceedings of the International Conference on World Wide Web*, WWW '16, pages 45–46, 2016. (Cited on page 84.)

[83] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. (Cited on pages 70 and 110.)

[84] C. Hong, I. Batal, and M. Hauskrecht. A mixtures-of-trees framework for multi-label classification. In *Proceedings of the ACM International Conference on Conference on Information and Knowledge*

Management, CIKM, pages 211–220, 2014. (Cited on page 15.)

[85] W.-N. Hsu, Y. Zhang, and J. Glass. Recurrent neural network encoder with attention for community question answering. *arXiv preprint arXiv:1603.07044*, 2016. (Cited on page 121.)

[86] J. Huang, G. Li, Q. Huang, and X. Wu. Learning label-specific features and class-dependent labels for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3309–3323, 2016. (Cited on page 15.)

[87] J. Huang, M. Peng, H. Wang, J. Cao, W. Gao, and X. Zhang. A probabilistic method for emerging topic tracking in microblog stream. *World Wide Web*, 20(2):325–350, 2017. (Cited on page 84.)

[88] M. Iivonen. Consistency in the selection of search concepts and search terms. *Information Processing & Management*, 31(2):173 – 190, 1995. (Cited on page 31.)

[89] M. Ioannou, G. Sakkas, G. Tsoumakas, and I. Vlahavas. Obtaining bipartitions from score vectors for multi-label classification. In *Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '10, pages 409–416, 2010. (Cited on page 16.)

[90] A. Jatowt and K. Duh. A framework for analyzing semantic change of words across time. In *Proceedings of the 14th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '14, pages 229–238, 2014. (Cited on pages 4, 81, 84, and 91.)

[91] G. Jeh and J. Widom. Simrank: A measure of structural-context similarity. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, SIGKDD '02, pages 538–543, 2002. (Cited on page 87.)

[92] J. Jeon, W. B. Croft, and J. H. Lee. Finding similar questions in large question and answer archives. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '05, pages 84–90, 2005. (Cited on page 119.)

[93] Z. Ji, F. Xu, B. Wang, and B. He. Question-answer topic model for question retrieval in community question answering. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '12, pages 2471–2474, 2012. (Cited on page 104.)

[94] J. Jiang and C. Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, ACL '07, pages 264–271, 2007. (Cited on page 62.)

[95] M. Jiang, Z. Pan, and N. Li. Multi-label text categorization using l21-norm minimization extreme learning machine. *Neurocomputing*, 261:4–10, 2017. (Cited on page 15.)

[96] V. Jijkoun and M. de Rijke. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '05, pages 76–83, 2005. (Cited on pages 104 and 119.)

[97] P. Jin, Y. Zhang, X. Chen, and Y. Xia. Bag-of-embeddings for text classification. In *Proceedings of the 25th International Conference on Artificial Intelligence*, IJCAI '16, pages 2824–2830, 2016. (Cited on page 85.)

[98] T. Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, SIGKDD '06, pages 217–226, 2006. (Cited on page 23.)

[99] Q. Ju, A. Moschitti, and R. Johansson. Learning to rank from structures in hierarchical text classification. In *Proceedings of the 35th European Conference on Advances in Information Retrieval*, ECIR '13, pages 183–194, 2013. (Cited on page 17.)

[100] C. Julie Gable and F. CDIA. Examining metadata: its role in e-discovery and the future of records managers. *Information Management*, 43(5):28, 2009. (Cited on page 1.)

[101] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014. (Cited on page 120.)

[102] A. K. Kalia, H. R. Motahari-Nezhad, C. Bartolini, and M. P. Singh. Monitoring commitments in people-driven service engagements. In *IEEE International Conference on Services Computing*, SCC '13, pages 160–167, 2013. (Cited on pages 61, 72, and 77.)

[103] T. Kenter and M. de Rijke. Attentive memory networks: Efficient machine reading for conversational search. In *1st International Workshop on Conversational Approaches to Information Retrieval*, 2017. (Cited on page 111.)

[104] T. Kenter, M. Wevers, P. Huijnen, and M. de Rijke. Ad hoc monitoring of vocabulary shifts over time. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '15, pages 1191–1200, 2015. (Cited on pages 4, 81, 82, 84, and 91.)

[105] T. Kenter, A. Borisov, and M. de Rijke. Siamese CBOW: Optimizing word embeddings for sentence representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL '16, pages 1411–1420, 2016. (Cited on page 120.)

[106] H. D. Kim and C. Zhai. Generating comparative summaries of contradictory opinions in text. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '09, pages 385–394, 2009. (Cited on page 85.)

[107] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *Proceedings of the European Conference on Machine Learning & Principles and Practice of Knowledge Discovery*, ECML-PKDD '04, pages 217–226, 2004. (Cited on pages 59 and 63.)

[108] E. Kotsakis. Structured information retrieval in xml documents. In *Proceedings of the symposium on Applied computing*, SAC '02, pages 663–667, 2002. (Cited on page 1.)

[109] W. M. Kouw, L. J. van der Maaten, J. H. Krijthe, and M. Loog. Feature-level domain adaptation. *Journal of Machine Learning Research*, 17(171):1–32, 2016. (Cited on page 62.)

[110] V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena. Statistically significant detection of linguistic change. In *Proceedings of the International Conference on World Wide Web*, WWW '15, pages 625–635, 2015. (Cited on pages 4, 81, 84, and 91.)

[111] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML'15, pages 957–966, 2015. (Cited on pages 84 and 85.)

[112] S. Lacoste-Julien, F. Sha, and M. I. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *Proceedings of Advances in Neural Information Processing Systems*, NIPS '09, pages 897–904, 2009. (Cited on page 36.)

[113] M. Lalmas and A. Trotman. XML retrieval. In *Encyclopedia of Database Systems*, pages 3616–3621. Springer, 2009. (Cited on page 1.)

[114] A. Lampert, C. Paris, and R. Dale. Can requests-for-action and commitments-to-act be reliably identified in email messages. In *Proceedings of the 12th Australasian Document Computing Symposium*, ADCS '07, pages 48–55, 2007. (Cited on pages 61 and 72.)

[115] A. Lampert, R. Dale, and C. Paris. Requests and commitments in email are more complex than you think: Eight reasons to be cautious. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 64–72, 2008. (Cited on pages 3, 59, 61, and 72.)

[116] A. Lampert, R. Dale, and C. Paris. Detecting emails containing requests for action. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '10, pages 984–992, 2010. (Cited on pages 3, 59, and 62.)

[117] G. Lample, A. Conneau, M. Ranzato, L. Denoyer, and H. Jegou. Word translation without parallel data. In *Proceedings of International Conference on Learning Representations*, ICLR '18, 2018. (Cited on page 62.)

[118] T. Lansdall-Welfare, S. Sudhahar, J. Thompson, J. Lewis, F. N. Team, and N. Cristianini. Content analysis of 150 years of british periodicals. *Proceedings of the National Academy of Sciences*, 114: 457–465, 2017. (Cited on pages 81 and 126.)

[119] J. H. Lau, D. Newman, and T. Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '14, pages 530–539, 2014. (Cited on pages 36, 39, and 44.)

[120] M. Law, M. Figueiredo, and A. Jain. Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166, 2004. (Cited on page 36.)

[121] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, ICML '14, pages 1188–1196, 2014. (Cited on page 92.)

[122] T. Lei, H. Joshi, R. Barzilay, T. Jaakkola, K. Tymoshenko, A. Moschitti, and L. Marquez. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL'16, pages 1279–1289, 2016. (Cited on page 104.)

[123] O. Levy and Y. Goldberg. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, CoNLL '14, pages 171–180, 2014. (Cited on page 84.)

[124] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004. (Cited on page 54.)

[125] B. Li, Q. Yang, and X. Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th International Conference on Machine Learning*, ICML '09, pages 617–624, 2009. (Cited on page 60.)

[126] C. Li, A. Sun, and A. Datta. Twevent: Segment-based event detection from tweets. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '12, pages 155–164, 2012. (Cited on page 84.)

[127] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '09, pages 375–384, 2009. (Cited on page 85.)

[128] C.-C. Lin, D. Kang, M. Gamon, M. Khabsa, A. H. Awadallah, and P. Pantel. Actionable email intent modeling with reparametrized rnns. *arXiv preprint arXiv:1712.09185*, 2017. (Cited on pages 3, 59, and 61.)

[129] T. Lin, W. Tian, Q. Mei, and H. Cheng. The dual-sparse topic model: Mining focused topics and focused terms in short text. In *Proceedings of the International Conference on World Wide Web*, WWW '14, pages 539–550, 2014. (Cited on pages 34, 35, and 40.)

[130] P. Liu, X. Qiu, and X. Huang. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL '17, pages 1–10, 2017. (Cited on pages 62 and 77.)

[131] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. (Cited on page 54.)

[132] G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49 (4):41–46, 2006. (Cited on page 1.)

[133] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie. Improving LDA topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 889–892, 2013. (Cited on pages 36 and 39.)

[134] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of the International Conference on World Wide Web*, WWW '07, pages 171–180, 2007. (Cited on page 85.)

[135] E. L. Mencía and J. Fürnkranz. Efficient multilabel classification algorithms for large-scale problems in the legal domain. In *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*, pages 192–215, 2010. (Cited on page 17.)

[136] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proceedings of International Conference on Learning Representations*, ICLR '13, 2013. (Cited on pages 84 and 90.)

[137] T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. In *Proceedings of International Conference on Learning Representations*, ICLR '13, 2013. (Cited on pages 62, 66, 82, and 86.)

[138] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems*, NIPS '13, pages 3111–3119, 2013. (Cited on pages 66, 91, and 120.)

[139] Z.-Y. Ming, T.-S. Chua, and G. Cong. Exploring domain-specific term weight in archived question search. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '10, pages 1605–1608, 2010. (Cited on pages 104 and 119.)

[140] G. Mishne and M. de Rijke. Source code retrieval using conceptual similarity. In *Coupling approaches, coupling media and coupling languages for information retrieval*, pages 539–554, 2004. (Cited on page 120.)

[141] E. Mohamed, M. Ehrmann, M. Turchi, and R. Steinberger. Multi label eurovoc classification for eastern and southern eu languages. In *Cambridge Scholars Publishing*, 2012. (Cited on page 18.)

[142] P. Nakov, D. Hoogeveen, L. Màrquez, A. Moschitti, H. Mubarak, T. Baldwin, and K. Verspoor. Semeval-2017 task 3: Community question answering. In *International Workshop on Semantic Evaluation*, SemEval' 17, pages 27–48, 2017. (Cited on pages 4, 103, 104, 105, 115, 119, and 121.)

[143] J. Nam, J. Kim, I. Gurevych, and J. Furnkranz. Large-scale multi-label text classification - revisiting neural networks. In *Proceedings of the European Conference on Machine Learning & Principles and Practice of Knowledge Discovery*, ECML-PKDD '14, pages 437–452, 2014. (Cited on pages 14, 15, and 16.)

[144] D. Newman, E. V. Bonilla, and W. Buntine. Improving topic coherence with regularized topic models. In *Proceedings of Advances in Neural Information Processing Systems*, NIPS '11, pages 496–504, 2011. (Cited on page 36.)

[145] H. R. M. Nezhad, K. Gunaratna, and J. Cappi. eassistant: Cognitive assistance for identification and auto-triage of actionable conversations. In *Proceedings of the International Conference on World Wide*

Web, WWW '17, pages 89–98, 2017. (Cited on pages 2, 3, and 59.)

[146] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313, 2015. (Cited on pages 36, 39, 44, and 54.)

[147] D. Oard, W. Webber, D. Kirsch, and S. Golitsynskiy. Avocado research email collection. *Linguistic Data Consortium*, 2015. (Cited on pages 59 and 63.)

[148] A. Olieman, H. Azarbonyad, M. Dehghani, J. Kamps, and M. Marx. Entity linking by focusing dbpedia candidate entities. In *SIGIR workshop on Entity Recognition and Disambiguation*, pages 13–24, 2014. (Cited on page 9.)

[149] K. D. Onal, Y. Zhang, I. S. Altingovde, M. M. Rahman, P. Karagoz, A. Braylan, B. Dang, H.-L. Chang, H. Kim, Q. McNamara, A. Angert, E. Banner, V. Khetan, T. McDonnell, A. T. Nguyen, D. Xu, B. C. Wallace, M. de Rijke, and M. Lease. Neural information retrieval: At the end of the early years. *Information Retrieval Journal*, 21(2–3):111–182, 2018. (Cited on pages 104 and 120.)

[150] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. (Cited on pages 60, 62, and 67.)

[151] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008. (Cited on page 85.)

[152] B. Patra. A survey of community question answering. *arXiv preprint arXiv:1705.04009*, 2017. (Cited on pages 4, 5, 103, 104, and 119.)

[153] S. Paul and A. Prakash. Querying source code using an algebraic query language. In *Proceedings of the International Conference on Software Maintenance*, ICSM '94, pages 127–136, 1994. (Cited on page 120.)

[154] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '14, pages 1532–1543, 2014. (Cited on page 84.)

[155] B. Pouliquen, R. Steinberger, and C. Ignat. Automatic annotation of multilingual text collections with a conceptual thesaurus. In *Proceedings of the Ontologies and Information Extraction Workshop*, EUROLAN, 2003. (Cited on pages 14 and 18.)

[156] Y.-P. Qin and X.-K. Wang. Study on multi-label text classification based on SVM. In *Proceedings of the Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, FSKD '09, pages 300–304, 2009. (Cited on page 15.)

[157] X. Qiu and X. Huang. Convolutional neural tensor network architecture for community-based question answering. In *Proceedings of the 25th International Conference on Artificial Intelligence*, IJCAI'15, pages 1305–1311, 2015. (Cited on page 120.)

[158] X. Qiu, W. Gao, and X. Huang. Hierarchical multi-class text categorization with global margin maximization. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACL '09, pages 165–168, 2009. (Cited on page 15.)

[159] J. R. Quevedo, O. Luaces, and A. Bahamonde. Multilabel classifiers with a probabilistic thresholding strategy. *Pattern Recognition*, 45(2):876–883, 2012. (Cited on page 16.)

[160] C. Quirk, P. Choudhury, J. Gao, H. Suzuki, K. Toutanova, M. Gamon, W.-t. Yih, L. Vanderwende, and C. Cherry. Msr splat, a language analysis toolkit. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '12, pages 21–24, 2012. (Cited on page 63.)

[161] R. Raina, A. Y. Ng, and D. Koller. Constructing informative priors using transfer learning. In *Proceedings of the 23rd International Conference on Machine learning*, ICML '06, pages 713–720, 2006. (Cited on page 60.)

[162] C. Rao. Diversity and dissimilarity coefficients: A unified approach. *Theoretical Population Biology*, 21(1):24–43, 1982. (Cited on pages 33 and 36.)

[163] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011. (Cited on page 16.)

[164] S. D. Reese and S. C. Lewis. Framing the war on terror the internalization of policy in the US press. *Journalism*, pages 777–797, 2009. (Cited on page 81.)

[165] A. Rehman, K. Javed, and H. A. Babri. Feature selection based on a normalized difference measure for text classification. *Information Processing & Management*, 53(2):473–489, 2017. (Cited on page 21.)

[166] Z. Ren, M.-H. Peetz, S. Liang, D. van Willemijn, and M. de Rijke. Hierarchical multi-label classification of social text streams. In *Proceedings of the 37th International ACM SIGIR conference on research and development in information retrieval*, SIGIR '14, pages 213–222, 2014. (Cited on page 14.)

[167] Z. Ren, O. Inel, L. Aroyo, and M. de Rijke. Time-aware multi-viewpoint summarization of multilingual social text streams. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '16, pages 387–396, 2016. (Cited on page 85.)

[168] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. (Cited on page 66.)

[169] M. Röder, A. Both, and A. Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 399–408, 2015. (Cited on page 44.)

[170] R. G. Rossi, A. de Andrade Lopes, and S. O. Rezende. Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts. *Information Processing & Management*, 52(2):217–257, 2016. (Cited on page 16.)

[171] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor. Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research*, 7:1601–1626, 2006. (Cited on pages 13 and 14.)

[172] H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000. (Cited on pages 60, 62, and 67.)

[173] A. A. Shiri, C. Revie, and G. Chowdhury. Thesaurus-enhanced search interfaces. *Journal of Information Science*, 28(2):111–122, 2002. (Cited on page 1.)

[174] A. Singh. Entity based Q&A retrieval. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1266–1277, 2012. (Cited on page 120.)

[175] Q. Skinner. Meaning and understanding in the history of ideas. *History and theory*, 8(1):3–53, 1969. (Cited on pages 81 and 126.)

[176] H. Soleimani and D. Miller. Parsimonious topic models with salient word discovery. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):824–837, 2015. (Cited on pages 34, 35, 36, 40, 41, 43, and 44.)

[177] A. Solow, S. Polasky, and J. Broadus. On the measurement of biological diversity. *Journal of Environmental Economics and Management*, 24(1):60–68, 1993. (Cited on page 33.)

[178] I. Srba and M. Bielikova. A comprehensive survey and classification of approaches for community question answering. *ACM Transactions on the Web*, 10(3):18:1–18:63, 2016. (Cited on pages 103, 119, and 121.)

[179] R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, and D. Tufis. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, LREC '06, 2006. (Cited on pages 14, 17, and 21.)

[180] R. Steinberger, M. Ebrahim, and M. Turchi. JRC EuroVoc indexer JEX-A freely available multi-label categorisation tool. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, LREC '12, 2012. (Cited on pages 14, 16, 18, and 21.)

[181] S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus, and R. Fergus. End-to-end memory networks. In *Proceedings of Advances in Neural Information Processing Systems*, NIPS '15, pages 2440–2448, 2015. (Cited on pages 121 and 126.)

[182] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proceedings of Advances in neural information processing systems*, NIPS '14, pages 3104–3112, 2014. (Cited on page 63.)

[183] H. Takamura, R. Nagata, and Y. Kawasaki. Analyzing semantic changes in japanese loanwords. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL '17, 2017. (Cited on page 84.)

[184] M. Tan, C. dos Santos, B. Xiang, and B. Zhou. LSTM-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*, 2015. (Cited on page 121.)

[185] L. Tang, S. Rajan, and V. K. Narayanan. Large scale multi-label classification via metalabeler. In *Proceedings of the International Conference on World Wide Web*, WWW '09, pages 211–220, 2009. (Cited on page 16.)

[186] T. Thonet, G. Cabanac, M. Boughanem, and K. Pinel-Sauvagnat. Vodum: A topic model unifying viewpoint, topic and opinion discovery. In *Proceedings of the 38th European Conference on Information Retrieval*, ECIR '16, pages 533–545, 2016. (Cited on page 85.)

[187] A. Triandafyllidou and R. Wodak. Conceptual and methodological questions in the study of collective identity: An introduction. *Journal of Language and Politics*, 2(2):205–223, 2003. (Cited on pages 81 and 126.)

[188] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685, 2010. (Cited on page 15.)

[189] G. Tur, A. Stolcke, L. Voss, S. Peters, D. Hakkani-Tur, J. Dowding, B. Favre, R. Fernandez, M. Frampton, M. Frandsen, C. Frederickson, M. Graciarena, D. Kintzing, K. Leveque, S. Mason, J. Niekrasz, M. Purver, K. Riedhammer, E. Shriberg, J. Tien, D. Vergyri, and F. Yang. The CALO meeting assistant system. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1601–1611, 2010. (Cited on pages 2, 3, and 59.)

[190] C. K. Vaca, A. Mantrach, A. Jaimes, and M. Saerens. A time-based collective factorization for topic discovery and monitoring in news. In *Proceedings of the International Conference on World Wide Web*, WWW '14, pages 527–538, 2014. (Cited on page 84.)

[191] A. van Aggelen, L. Hollink, and J. van Ossenbruggen. Combining distributional semantics and structured data to study lexical change. In *Proceedings of the 20th International Conference on Knowledge Engineering and Knowledge Management*, EKAW 16, pages 40–49, 2017. (Cited on page 84.)

[192] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems*, NIPS '17, pages 6000–6010, 2017. (Cited on page 108.)

[193] S. Verberne, E. Dahondt, A. van den Bosch, and M. Marx. Automatic thematic classification of election manifestos. *Information Processing & Management*, 50(4):554–567, 2014. (Cited on page 17.)

[194] D. Vilar, M. Castro, and E. Sanchis. Multi-label text classification using multinomial models. In *Proceedings of the 4th International Conference on Natural Language Processing*, NATP '04, pages 220–230, 2004. (Cited on page 15.)

[195] H. M. Wallach, D. M. Mimno, and A. McCallum. Rethinking lda: Why priors matter. In *Proceedings of Advances in Neural Information Processing Systems*, NIPS '09, pages 1973–1981, 2009. (Cited on pages 34, 35, and 40.)

[196] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, 2009. (Cited on page 36.)

[197] B. Wang and J. Tsotsos. Dynamic label propagation for semi-supervised multi-class multi-label classification. *Pattern Recognition*, 52:75 – 84, 2016. (Cited on page 16.)

[198] J. Wang, Z. Li, X. Hu, and B. Hu. A novel composite kernel for finding similar questions in CQA services. In *International Conference on Web-Age Information Management*, pages 608–619, 2010. (Cited on page 120.)

[199] K. Wang, Z. Ming, and T.-S. Chua. A syntactic tree matching approach to finding similar questions in community-based QA services. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 187–194, 2009. (Cited on page 35.)

[200] K. Wang, Z.-Y. Ming, X. Hu, and T.-S. Chua. Segmentation of multi-sentence questions: Towards effective question retrieval in cQA services. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 387–394, 2010. (Cited on page 120.)

[201] P. Wang, Y. Zhang, L. Ji, J. Yan, and L. Jin. Concept embedded convolutional semantic model for question retrieval. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 395–403, 2017. (Cited on page 120.)

[202] Z. Wang, W. Hamza, and R. Florian. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 27th International Conference on Artificial Intelligence*, IJCAI'17, pages 4144–4150, 2017. (Cited on page 114.)

[203] J. Wehrmann, R. C. Barros, S. N. d. Dôres, and R. Cerri. Hierarchical multi-label classification with chained neural networks. In *Proceedings of the Symposium on Applied Computing*, SAC '17, pages 790–795, 2017. (Cited on page 16.)

[204] J. Weston, S. Chopra, and A. Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014. (Cited on pages 108 and 121.)

[205] R. W. White and R. A. Roth. Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–98, 2009. (Cited on page 1.)

[206] R. W. White, B. Kules, S. M. Drucker, et al. Supporting exploratory search, introduction, special issue, communications of the acm. *Communications of the ACM*, 49(4):36–39, 2006. (Cited on page 1.)

[207] S. Whittaker and C. Sidner. Email overload: Exploring personal information management of email. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, SIGCHI '96, pages

276–283, 1996. (Cited on pages 2, 3, 59, and 61.)

[208] R. Wilkinson. Effective retrieval of structured documents. In *Proceedings of the 17th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 94, pages 311–317, 1994. (Cited on page 1.)

[209] S. Williamson, C. Wang, K. A. Heller, and D. M. Blei. The IBP compound Dirichlet process and its application to focused topic modeling. In *Proceedings of the 27th International Conference on Machine Learning*, ICML '10, pages 1151–1158, 2010. (Cited on page 35.)

[210] L. Wittgenstein. Philosophische Untersuchungen (Frankfurt am Main: Suhrkamp). 1967. (Cited on pages 81 and 126.)

[211] H. Wu, W. Wu, M. Zhou, E. Chen, L. Duan, and H.-Y. Shum. Improving search relevance for short queries in community question answering. In *Proceedings of the Seventh ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 43–52, 2014. (Cited on pages 104 and 120.)

[212] P. Wu and T. G. Dietterich. Improving SVM accuracy by training on auxiliary data sources. In *Proceedings of the 21st International Conference on Machine learning*, ICML '04, page 110, 2004. (Cited on page 60.)

[213] P. Xie and E. P. Xing. Integrating document clustering and topic modeling. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 694–703, 2013. (Cited on page 36.)

[214] C. Xing, D. Wang, X. Zhang, and C. Liu. Document classification with distributions of word vectors. In *Procceedings of Signal and Information Processing Association Annual Summit and Conference*, APSIPA '14, pages 1–5, 2014. (Cited on page 85.)

[215] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. In *Proceedings of the 33rd International Conference on Machine Learning*, ICML'16, pages 2397–2406, 2016. (Cited on page 111.)

[216] J. Xu and H. Li. Adarank: A boosting algorithm for information retrieval. In *Proceedings of the 30th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 391–398, 2007. (Cited on pages 18 and 21.)

[217] X. Xue, J. Jeon, and W. B. Croft. Retrieval models for question and answer archives. In *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 475–482, 2008. (Cited on page 120.)

[218] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. In *Proceedings of the International Conference on World Wide Web*, WWW '13, pages 1445–1456, 2013. (Cited on page 36.)

[219] Y. Yang and S. Gopal. Multilabel classification with meta-level features in a learning-to-rank framework. *Machine Learning*, 88(1-2):47–68, 2012. (Cited on pages 13, 14, 16, 17, 26, and 27.)

[220] Z. Yao, Y. Sun, W. Ding, N. Rao, and H. Xiong. Discovery of evolving semantics through dynamic word embedding learning. *arXiv preprint arXiv:1703.00607*, 2017. (Cited on page 84.)

[221] P. Yuan, Y. Chen, H. Jin, and L. Huang. MSVM-kNN: Combining SVM and k-NN for multi-class text classification. In *Proceedings of the IEEE International Workshop on Semantic Computing and Systems*, WSCS '08, pages 133–140, 2008. (Cited on page 15.)

[222] H. Zamani and W. B. Croft. Embedding-based query language models. In *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ICTIR '16, pages 147–156, 2016. (Cited on page 120.)

[223] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the International Conference on Information and Knowledge Management*, CIKM '01, pages 403–410, 2001. (Cited on page 37.)

[224] K. Zhang, W. Wu, H. Wu, Z. Li, and M. Zhou. Question retrieval with high quality answers in community question answering. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '14, pages 371–380, 2014. (Cited on page 121.)

[225] K. Zhang, W. Wu, F. Wang, M. Zhou, and Z. Li. Learning distributed representations of data in community question answering for question retrieval. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 533–542, 2016. (Cited on pages 4, 5, 103, 115, and 121.)

[226] M.-L. Zhang and Z.-H. Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transaction on Knowledge and Data Engineering*, 18(10), 2006. (Cited on page 16.)

[227] M. L. Zhang and Z. H. Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014. (Cited on pages 15 and 16.)

[228] Y. Zhang, J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, 2002. (Cited on page 37.)

[229] G. Zhou, Y. Chen, D. Zeng, and J. Zhao. Towards faster and better retrieval models for question search. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, CIKM '13, pages 2139–2148, 2013. (Cited on pages 104 and 120.)

[230] G. Zhou, Y. Liu, F. Liu, D. Zeng, and J. Zhao. Improving question retrieval in community question answering using world knowledge. In *Proceedings of the 23rd International Conference on Artificial Intelligence*, IJCAI '13, pages 2239–2245, 2013. (Cited on page 120.)

[231] G. Zhou, T. He, J. Zhao, and P. Hu. Learning continuous word embedding with metadata for question retrieval in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, ACL '15, pages 250–259, 2015. (Cited on pages 4, 5, 103, 104, 115, 119, and 120.)

[232] G. Zhou, Y. Zhou, T. He, and W. Wu. Learning semantic representation with neural networks for community question answering retrieval. *Know.-Based Syst.*, 93(C):75–83, 2016. (Cited on page 120.)

[233] F. Zhuang, P. Luo, H. Xiong, Y. Xiong, Q. He, and Z. Shi. Cross-domain learning from multiple sources: A consensus regularization perspective. *IEEE Transactions on Knowledge and Data Engineering*, 22 (12):1664–1678, 2010. (Cited on page 62.)

[234] F. Zhuang, X. Cheng, P. Luo, S. J. Pan, and Q. He. Supervised representation learning: Transfer learning with deep autoencoders. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI '15, pages 4119–4125, 2015. (Cited on page 62.)

Summary

The Web is a major source of information for many users. Automatically grouping and classifying documents on the Web is an important ingredient when trying to support effective search of Web documents. Documents on the Web are a mixture of content (text, image, etc.), structure, and metadata. Beside content, metadata and structure can be helpful in managing documents and supporting exploratory search over them. In this thesis, we aim at empowering content-based approaches—for managing documents and exploratory search—with structure and metadata and study how metadata and structure associated with documents can help to both manage documents more accurately and support different exploratory search tasks.

In the first part of the thesis, we study how we can leverage metadata and structure to manage documents. We define document management as the task of grouping similar documents together and classifying them. We define the similarity from three different angles in the three chapters of the first part. We first start by classifying documents based on their topical similarity and use content, structure, and metadata associated with documents to classify them. We show that a metadata/structure powered classifier performs much better than a classifier that is solely based on the content. We then focus on classifying documents based on their topical diversity. We measure topical diversity by means of topic models. In doing this, we characterize the main drawbacks of topic models when they are applied to the task of measuring topical diversity and propose a hierarchical approach to address these drawbacks. We find that the topic models achieved using our hierarchical approach not only have a superior performance in the topical diversity task compared to the previous approaches, but also are useful in other tasks such as classification and clustering of topically similar documents. Finally, we move to the email domain and study how we can help users automatically manage their tasks created via email. We focus on the task of detecting commitments made in email as it enables digital assistants to help their users recall promises they have made and assist them in meeting those promises in a timely manner. We show that domain bias associated with email corpora has a large negative impact on the performance of commitment detection models. We adapt and use transfer learning methods to remove the domain bias from email corpora and show that by means of transfer learning we can reliably detect commitments even if there is a domain bias.

In the second part of the thesis, we devote two chapters to study how metadata and structure can be useful in two different exploratory search tasks. First, we consider detecting shifts in the meaning of words and study how metadata features such as time and social entities such as political party can be used to discover shifts in a short period of time. We use distributional semantics models to represent the meaning of words and propose different approaches to measure the difference between the meaning of a term in different semantic spaces. We show that the detected shifts can be helpful in various tasks such as automatic text summarization and ideology detection. Second, we focus on finding similar questions in community question answering forums. Our aim is to use structure and metadata associated with questions beside their content to measure the semantic similarity between questions. We show that a neural model that effectively exploits the heterogeneous information associated with questions can improve the performance of content-based approaches in this task.

Samenvatting

Het web is voor veel gebruikers een belangrijke informatiebron. Het automatisch groeperen en classificeren van documenten op het web is een belangrijk ingrediënt voor het effectief zoeken naar webdocumenten. Documenten op het web bestaan uit een combinatie van verschillende soorten inhoud (tekstueel, visueel, etc.), structuur en metadata. Naast inhoud, kunnen ook metadata en structuur nuttig zijn bij het beheren van en verkennend zoeken naar webdocumenten. Ons doel in dit proefschrift is het versterken van inhoud gebaseerde methodes—voor het beheren van en zoeken naar documenten—met structuur en metadata en het bestuderen hoe metadata en de structuur van documenten kunnen helpen om zowel documenten accurater te beheren en verschillende verkennende zoektaken beter te ondersteunen.

In het eerste deel van dit proefschrift, bestuderen we hoe we het best gebruik kunnen maken van de metadata en structuur om documenten te beheren. We definiëren documentbeheer als de taak om soortgelijke documenten samen te groeperen en te classificeren. We definiëren gelijksoortigheid van documenten vanuit drie verschillende perspectieven in de eerste drie hoofdstukken. We beginnen met het classificeren van documenten op basis van hun *topical similarity*, de overeenkomst in het onderwerp van de documenten, en gebruiken hiervoor zowel de inhoud, structuur als metadata van de documenten. We laten zien dat een classifier dat gebruik maakt van de extra metadata/structuur beter werkt dan een classifier dat enkel gebaseerd is op inhoud. Vervolgens concentreren we ons op het classificeren van documenten op basis van *topical diversity*, de diversiteit in onderwerpen van een document. We meten de diversiteit door middel van *topic models*. Daarmee kenmerken we de belangrijkste nadelen van het gebruik van *topic models* voor het meten van diversiteit en introduceren wij een hiërarchische aanpak om deze nadelen aan te pakken. Als uitkomst vinden we dat een *topic model* met een hiërarchische aanpak niet alleen voordelen heeft voor het vinden van diverse onderwerpen binnen documenten, maar ook bruikbaar is voor andere taken zoals document classificatie en het clusteren van soortgelijke documenten. Ten slotte gaan we naar het domein van e-mails en bestuderen we hoe we gebruikers kunnen helpen hun taken, die via e-mail zijn gemaakt, automatisch te beheren. We richten ons op het detecteren van toezeggingen in e-mails, omdat dit digitale assistenten in staat stelt gebruikers van hun beloften te herinneren en hen tijdig te helpen bij het nakomen van deze beloften. E-mail corpora worden geassocieerd met een bepaald domein. Dit wordt de *domain bias* genoemd. We tonen aan dat deze *domain bias* er voor zorgt dat toezeggingen die gedaan worden in e-mails minder goed gedetecteerd worden. We passen *transfer learning* methodes aan en gebruiken deze om de *domain bias* van e-mailcorpora te verwijderen en tonen aan dat we door middel van *transfer learning* betrouwbaar toezeggingen kunnen detecteren, zelfs wanneer er sprake is van *domain bias*.

In het tweede deel van dit proefschrift, wijden we twee hoofdstukken toe aan de studie van hoe metadata en structuur ingezet kunnen worden voor twee verschillende verkennende zoektaken. Ten eerste beschouwen we het ontdekken van verschuivingen in de betekenis van woorden en bestuderen we hoe metadata zoals tijd en sociale entiteiten zoals de politieke partij gebruikt kunnen worden om deze verschuivingen in een korte periode te ontdekken. We maken gebruik van *distributional semantics models* om de

betekenis van woorden te modelleren en stellen verschillende methodes voor om het verschil in betekenis van een woord te kunnen meten in verschillende semantische ruimtes. Wij laten zien dat het detecteren van zulke verschuivingen nuttig kan zijn voor verschillende taken zoals het automatisch samenvatten van tekst en het ontdekken van ideologie. Ten tweede concentreren we ons op het vinden van soortgelijke vragen op vraagbeantwoording forums. Ons doel is om, naast de inhoud van de vraag, ook de structuur en metadata geassocieerd met de vraag te gebruiken om soortgelijke vragen te vinden. We laten zien dat een neuraal model, dat effectief de heterogene informatie geassocieerd met vragen gebruikt, beter werkt dan de inhoud-gebaseerde aanpakken voor deze taak.