Information Preparation with the Human in the Loop

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erlangung des akademischen Grades
Doktor-Ingenieur

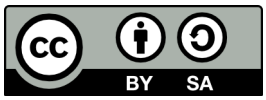
vorgelegt von

Avinesh P.V.S., M. Sc.
geboren in Bellampally, India

Tag der Einreichung: 25. April 2019
Tag der Disputation: 18. Juli 2019

Referenten: Prof. Dr. phil. Iryna Gurevych, Darmstadt
Prof. Mark Sanderson, Melbourne

Darmstadt 2019
D17



Abstract

With the advent of the World Wide Web (WWW) and the rise of digital media consumption, abundant information is available nowadays for any topic. But these days users often suffer from information overload posing a great challenge for finding relevant and important information. To alleviate this information overload and provide a significant value to the users, there is a need for automatic information preparation methods. Such methods need to support users by discovering and recommending important information while filtering redundant and irrelevant information. They need to ensure that the users do not drown in, but rather benefit from the prepared information. However, the definition of what is relevant and important is subjective and highly specific to the user's information need and the task at hand. Therefore, a method must continually learn from the feedback of its users. In this thesis, we propose new approaches to put the human in the loop in order to interactively prepare information along the three major lines of research: information aggregation, condensation, and recommendation.

For multiple well-studied tasks in natural language processing, we point out the limitation of existing methods and discuss how our approach can successfully close the gap to the human upper bound by considering user feedback and adapting to the user's information need. We put a particular focus on applications in digital journalism and introduce the new task of live blog summarization. We show that the corpora we create for this task are highly heterogeneous as compared to the standard summarization datasets which poses new challenges to previously proposed non-interactive methods.

One way to alleviate information overload is information aggregation. We focus on the corresponding task of multi-document summarization and argue that previous proposed methods are of limited usefulness in real-world application as they do not take the users' goal into account. To address these drawbacks, we propose an interactive summarization loop to iteratively create and refine multi-document summaries based on the users' feedback. We investigate sampling strategies based on active machine learning and joint optimization to reduce the number of iterations and the amount of user feedback required. Our approach significantly improves the quality of the summaries and reaches a performance near the human

upper bound. We present a system demonstration implementing the interactive summarization loop, study its scalability, and highlight its use cases in exploring document collections and creating focused summaries in journalism.

For information condensation, we investigate a text compression setup. We address the problem of neural models requiring huge amounts of training data and propose a new interactive text compression method to reduce the need for large-scale annotated data. We employ state-of-the-art Seq2Seq text compression methods as our base models and propose an active learning setup with multiple sampling strategies to efficiently use minimal training data. We find that our method significantly reduces the amount of data needed to train and that it adapts well to new datasets and domains.

We finally focus on information recommendation and discuss the need for explainable models in machine learning. We propose a new joint recommendation system of rating prediction and review summarization, which shows major improvements over state-of-the-art systems in both the rating prediction and the review summarization task. By solving this task jointly based on multi-task learning techniques, we furthermore obtain explanations for a rating by showing the generated review summary marked based on the model's attention and a histogram of user preferences learned from the reviews of the users.

We conclude the thesis with a summary of how human-in-the-loop approaches improve information preparation systems and envision the use of interactive machine learning methods also for other areas of natural language processing.

Acknowledgements

Ph.D. has been a life-changing journey and I want to take this opportunity to give a shout out to all those without whom this journey would not have been a memorable one.

Firstly, I would like to express my deepest gratitude to Dr. Christian M. Meyer for giving me the opportunity to pursue research, for supporting me throughout my Ph.D. journey by spending a significant amount of time reading, improving and providing useful feedback to my drafts, and also helped in developing my writing skills. Thanks for being a great supervisor and a role model. I hope to be like you one day. I am also grateful for my second supervisor Prof. Iryna Gurevych for accepting me as a Ph.D. candidate, supporting me during the initial stages and providing me useful pointers to carry forward my research. I would further like to thank Prof. Mark Sanderson for inviting me to RMIT Melbourne for a research visit during my Ph.D. and kindly agreeing to be a reviewer of this thesis. Like my acknowledgments at the end of every paper, my sincere appreciation goes to DFG for generously funding this Ph.D. research through the "Adaptive Preparation of Information from Heterogeneous Sources" (AIPHES, GRK 1994/1) research training group.

AIPHES played an essential role in creating a collaborative research environment. In particular, I am thankful to Andreas, Christopher, Chinnappa, Gerold, Markus, Maxime, Teresa, Thomas, Tobias, Sergey, Ana, Todor and Benjamin for being supportive colleagues. Thanks for numerous informal and formal discussions during our lunch at the cafeteria and the Friday coffee sessions. And also I want to thank the new cohorts, Aicha, Swetha, Wei, Leonardo, Aiji, Benjamin for fruitful collaborations and discussions on various topics. I am grateful to the UKP Lab colleagues for the knowledge sharing events and useful feedback on presentations, papers during our weekly SIG group, UKP meetings. Special thanks to Alex, Yevgeniy, Edwin, JiUng, Daniil, Edwin, Nils, Ivan and many other current and former members. Thanks for all the support and friendships throughout the last years.

During my Ph.D., as a research visit, I managed to escape one European winter by visiting Down Under. Thanks to Dr. Yongli Ren and Prof. Mark Sanderson for inviting me and introducing me to recommendation systems, which was one problem I always wanted to work. Thanks to Eliezer, Jan for being amazing international colleagues at RMIT, Melbourne. I will

miss the formal and informal discussions I had with you. I have greatly benefited from my stay at RMIT, and I also want to thank Jeff and Zhifeng for being fabulous collaborators. A special thanks to Shridhar, Sidhant for being exceptional flatmates and friends for life, I could not have asked for more during my time at Melbourne.

I was fortunate to do an internship at Apple, Siri Cloud Services Localization team in Cupertino, one of the leading teams providing natural language solutions to various languages across the globe. I worked with Siddharth Patwardhan my IBuddy and mentor, Sachin Agarwal the coolest manager and great teammates thanks to Alex, Ravikiran, Eric, Shaona, Pallavi, Deepanshu, Jean-Philippe, Huiting, Yinying, Jordan. This internship paved my way to head back to the industry to a full-time position. I would also like to express my deepest gratitude to Madhav chinnananna, Kiran chinnamma, Vandana akka and Vasu bavagaru for being amazing hosts in the USA during my three months. Caio, thanks for being a close friend, flatmate, gym buddy and for all those valuable discussions, fun Nintendo sessions and many more during the internship.

Last but not least, I am in debt to my friends and family. I have always been a rebellious son to my parents. They have always supported me all through my life with every single choice. Thanks to my dearest brother and sister-in-law for being the support of my family. I am always grateful for having great friends Saladi Rahul, Ravikiran, Siva Reddy, Spandana, Bharat Ambati, Abhilash who have always supported me personally and professionally. Thanks to Prof. Sangal, Prof. Dipti Ma'm, Adam, Diana and Simon for being my first guides to research and for always encouraging me to achieve great heights, I consider you all more as a family. Adam, your loss left a hole in all our lives, and I am forever in debt for the opportunity that you gave us to do research.

Thanks to my friends in Darmstadt: Julian for lifelong friendship and motivations, Yanai for being the enthusiastic guy and introducing me to half of my friends in Darmstadt, Andi for our intellectual Sunday lunch sessions, Hari, Aakash, Sergey for everlasting friendships and fun times, Tamisra for spontaneous Indian food. Radhika, Ranjani, Varsha, Ankush, Tushar for badminton, travel, and party, Caroline, Kazim for fun and fitness, Ramjee, Sylvia, Gerard for meditation sessions and many more. Thanks all for making my stay in Germany a memorable one. This journey would be incomplete without the two special Spanish ladies, Isabelle my dearest Spanish mom and Tami, my Spanish teacher/friend. Thanks for being family away from home and always sending warmth for my encouragement.

Contents

Chapter 1

Introduction

"Information is abundant, it flows through so many sources that what once was a river one waded through is now a flood we struggle to keep afloat in."
— Aysha Taryam

In recent decades, with the advent of the information age, for the first time, people could access information easily with the click of a button. Vast volumes of new data are continuously created through websites, news articles, blogs, radio, television, print media, e-mail, RSS feeds, etc. A recent study showed that 90% of all the data in the world is created during the last two years. [1] Over 2.5 Quintillion bytes of data are created every day and this is accelerating with the ubiquitous use of new technologies, user devices and the Internet of Things.[2] Moreover, according to a study conducted by the International Data Corporation on a worldwide basis, these volumes of data are increasing at a rapid pace, and by 2020, the amount of information on the web will increase to 44 zettabytes or trillion gigabytes (Gantz and Reinsel, 2013).

However, these vast volumes of information also cause *information overload* (Patterson et al., 2001; Keim et al., 2008; Allen and Wilson, 2003), which has become a major problem in today's world. Information overload occurs when the abundant information available exceeds the processing capacity of humans. Today's digital media user receives a vast number of information bits every moment and their cognitive ability is unable to process it. Even if we had superhuman reading speeds and could retain vast amounts of information, it is physically impossible to digest all the information on any topic. The cause of this information overload is the rapid rate of creating information on the WWW using the digital media. Furthermore, digital devices like computers, smartphones, tablets acts as an easy medium to diffuse information with the help of social media, which causes the danger of drowning in a flood of information. In the recent work, Hoq (2016) pointed out that information overload is usually

[1] https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/ last accessed on 19^{th} March 2019

[2] https://www.domo.com/solution/data-never-sleeps-6 last accessed on 19^{th} March 2019

caused due to the following reasons: (a) by the existence of multiple sources of information, (b) over-abundance of information, (c) difficulty in managing information, (d) irrelevance of received information and (e) lack of user's time to analyze and understand information. Although the information overload and the causes are applicable to various domains, in this thesis, we particularly address this from a journalistic perspective.

1.1 Information Overload in Journalism

Information overload is a particular problem in the area of journalism concerning both reporting about news events and consuming the vast amount of content (Holton and Chyi, 2012). On the one hand, journalists are overwhelmed by the amount of information they are required to process every day. Journalists are always fighting to keep ahead in terms of delivering updates to their readers. The breaking news is broadcasted and spun across 24/7 news broadcasters, radio stations, social media platforms. To address this issue, journalists actively seek for new technologies to ease the preparation of information with automated means.

One such example was the analysis of the Panama Paper leak in 2016 investigated by the ICIJ. The data from the Panama Papers consist of 2.6 terabytes in 11.5 billion documents.[3] Technology played a vital role where a network of more than 100 journalists collaborated and reported the news story around the world. ICIJ specialists created a secure chat platform where every journalist contributed in real time with leads found in the data. Panama Papers is one such use case where technology transformed the landscape of newsrooms to deal with the exponential growth of information. Other journalistic use cases include tools for automatic fake news detection, fact checking, toxic comment classification, crowdsourcing-based information gathering, social media analytics for news, propagandistic style detection, trend prediction, news summarization , etc. As never before, there is a growing need for the journalistic tools to ease the process of information preparation which adapts to the journalist's requirements.

On the other hand, the existence of numerous media organizations and the rise of social media inflates the information available to a reader (York, 2013). The amount of time that a reader has to read all the news is limited and the ease of digesting information that specially interest the reader, becomes harder and harder.[4] Thus, readers are also exploring solutions provided by third-party application developers to digest the 24/7 news content. Popular news applications like Yahoo News Digest, Google Reader, Google News, Flipboard, Feedly, News 360, Apple News, SmartNews , etc., provide aggregated personalized news information to the user's device. A user first creates a personalized profile according to their preferences and then automatic content aggregators disseminate the relevant content based on those preferences. There are also apps like Pocket and Pinterest to stash content to be read later. Additionally,

[3] https://www.icij.org/investigations/panama-papers/ last accessed on 19^{th} March 2019

[4] http://openaccess.city.ac.uk/16131/ last accessed on 11^{th} April 2019

there are apps like Reddit and Digg which aim to bring popular content for the user, using proprietary algorithms and crowdsourcing-based human curation. Although there are various applications for information preparation, they are mostly static for the users needs and cannot fulfill all user requirements. Thus, tools are needed to enable a user to model the application to adapt to the user's needs in an efficient and effective way.

1.2 Alleviate Information Overload

Information is only as useful as the amount of knowledge we can derive from it. Intelligent ways to alleviate information overload by means of information preparation can generate significant value to the users. An important question thus becomes how do we help users avoid information overload such that we can reduce the burden of information? Information overload can be avoided by intelligently preparing information which involves the following activities (Pollar, 2004):

- *Discover*: Exploring new information and understanding the "big picture" i.e., the broadness or structure of all the information in a certain context. For example, things like get-an-overview (e.g., by bullet points), assist a user in exploring and summarizing important information (Over and Yen, 2001; Dang and Owczarzak, 2008).
- *Filter*: The less information presented, the easier it is to understand for the user and less likely to make them overwhelmed. For example, creating filters for e-mails ensures that spam mail does not catch our attention (Cormack, 2008). Similarly, in a journalistic use case, filtering out unimportant tweets or events while following a live event like elections helps the reader (Kelly, 2009), automatically creating compressed headlines by filtering unimportant information (Filippova and Altun, 2013) helps the journalist.
- *Adapt*: If easing a user's information need is the goal, then we need to make the information accessible to the user and make it obvious how to customize. For example, suggesting the user news articles based on their reading preferences (Liu et al., 2010) and letting the user to customize the selection based on feedback (Eppler and Mengis, 2004).

In this thesis, we study methods for information preparation to alleviate information overload. Journalistic consumer needs can be broadly divided into information summarization, information condensation, and information recommendation. Most of the past research in the areas of information summarization (Boudin et al., 2015), information condensation (Filippova and Altun, 2013), and information recommendation (Zheng et al., 2017) has primarily focused on building generic systems, however, they have missed to capture the users' needs like personalization, customization and explanations. For this, we need methods to support users by discovering and recommending important information while filtering redundant and

irrelevant information. However, the definition of what is relevant and important is subjective and highly specific to task at hand and the user's information need. Therefore, a method must continually learn with feedback using human in the loop. In this thesis, we propose new approaches to put the human in the loop in order to interactively prepare information along the three major lines of research: information summarization, condensation, and recommendation. In an *information summarization* scenario, we distinguish to discover important information (e.g. text) from potentially many heterogeneous sources in order to facilitate a user's information need. After identifying the relevant content, the natural next step is *information condensation*, which is to condense information by filtering unimportant or redundant information. Finally, in the *information recommendation* scenario, we learn to understand what a user needs or is interested in over time and recommend items (e.g. news articles). In Chapter 2, we provide a detailed definition of the three scenarios and relevant applications to alleviate information overload in journalism.

The goal of this thesis is to research new methods for information preparation that put the human in the loop and adapt to their needs. Our research is guided by the following research questions:

(1) How do the state-of-the-art fully automatic methods perform in information preparation, specifically, information summarization, information condensation, and information recommendation?
(2) How can we use human-in-the-loop such that information preparation adapts to the users' needs?
(3) How can we have time and data efficient learning in information preparation with the human-in-the-loop?
(4) How can we use user preferences to explain information preparation?

1.3 Contributions

In order to answer the research questions, we present new human in the loop methods and present comprehensive experiments resulting in a new state-of-the-art for information summarization, information condensation and information recommendation. The following is the summary of the contributions we make in the thesis:

- **Contribution 1: We introduce information preparation in journalism.**
  We analyze information preparation in journalism along the three lines of research: information summarization, condensation, and recommendation. To this end, we choose multi-document summarization, text compression, and item recommendation as prototypical application scenarios. We show how existing solutions for information preparation can be transferred to journalism. We also introduce a novel task in information summarization of automatic live blog summarization which has direct applications

in digital journalism and news research. With regards to live blog summarization, we propose a corpus construction approach to collect and extract live blogs with human-written summaries. The tools and methods to construct such corpora for two major online newspapers are made publicly available. Furthermore, we show that our research not only suits homogeneous data, but also works for heterogeneous data sources. Lastly, we experiment with multiple languages and our methods can be transferred to a new language with a minimal effort.

- **Contribution 2: We propose new human-in-the-loop approaches for information preparation**.
  We propose a novel interactive approach to identify important content in creating multi-document summaries that capture the user's demands. To solve the challenges of the current state-of-the-art systems, we propose pool-based active learning and joint optimization techniques to minimize the amount of user feedback required for identifying important concepts. Finally, we benchmark standard reference datasets and our novel live blog summarization corpus with the commonly used state-of-the-art summarization methods and empirically compare with our proposed model.

- **Contribution 3: We propose new time-efficient and data-efficient interactive models**.
  We implement Sherlock: an interactive summarization system for large text collections. In this collaboration work, we propose a new approximate summarization model to achieve interactive speeds and keep the users engaged in the process of creating summaries for large document collection. Sherlock, as an application, demonstrates how it can be used for query-focused summarization and efficiently browsing large document collections. to address the data efficiency bottleneck, we propose new interactive models for neural information compression in the area of text condensation. Our proposed models solve the need for large training data for neural sequence-to-sequence models. Furthermore, these models can be directly used for data-efficient training as part of the annotation workflow. We propose new sampling strategies to select the samples with high uncertainty for training in an interactive setup. We empirically compare our proposed sampling strategies with random sampling using state-of-the-art neural text compression methods for both in-domain training as well as domain adaptation.

- **Contribution 4: We propose new explainable models using user preferences**.
  Explanation-based recommendation systems received little attention in previous work. We propose an explainable recommendation system that generates user-adapted summaries to explain the system's recommendation. We propose a novel joint learning model that jointly learns to predict the possible rating for an item and summarizes the content based on user preferences and is the state-of-the-art system for the product rec-

ommendation. The proposed model includes the aspect of transparency of decisions of item recommendations. Such a system can be used for recommending news articles to the readers for a journalistic use case.

1.4 Publication Record

The majority of the contributions of this thesis have been previously published at peer-reviewed conferences in the fields of computational linguistics, NLP, information management, and computer science. In the following paragraphs, we describe these publications and link the chapters of the thesis to the contributions on which they are based, including verbatim of the quoted text.

The live blog summarization corpus is first developed in P.V.S. et al. (2018b). The publication introduces a method to create a new benchmark live blog summarization corpus and discusses the limitations of the state-of-the-art systems for this task. The contents of the work are partly used in Chapter 3, in particular Section 3.2 and to a large extent in Section 4.3.2 of Chapter 4.

The publication by P.V.S. and Meyer (2017) focuses on our interactive summarization framework to address the limitations of the current state-of-the-art systems. In Section 4.4 of this thesis, we incorporate the methodology described in this publication to efficiently produce high-quality summaries with a minimum number of iterations and feedback. In addition, we add parts of the analysis from it to Section 4.6. In Section 4.7, the demonstration of this interactive summarization system Sherlock is published in P.V.S. et al. (2018a). The publication contains information approximation measures taken to scale the interactive summarization to achieve interactive speeds. Section 4.6.5 of this thesis is based on it.

Additionally, closely connected to information summarization is the work by Zopf et al. (2018). In a joint publication with other researchers from AIPHES, we investigate the usefulness of different linguistic annotations for identifying content importance in automatic summarization. The annotations contributed to this work are the content phrases extracted from the text and the techniques used to obtain these content phrases are part of the Chapter 4. However, the content of this publication is not part of the thesis.

In the area of information condensation we published our data-efficient methods for text compression in P.V.S. and Meyer (2019). The paper contains a description of the current state-of-the-art text compression methods and our proposed novel interactive text compression framework. Additionally, the paper also describes the experimental comparisons of our newly proposed data sampling techniques with the state-of-the-art systems. The content of the paper, together with experimental results and analysis, is the basis of Chapter 5. Passages of this publication are quoted verbatim.

Lastly, in the area of information recommendation, we published our methods for item recommendation in P.V.S. et al. (2019). In Section 6.3 of this thesis, we incorporate our proposed novel joint rating prediction, review summarization based recommendation system method-

ology described in this publication, and compare it with the current state-of-the-art recommendation methods. Additionally, the paper also showcases ways of explaining the predicted rating. Closely related to the topic of explaining the system's recommendations and journalism is explaining a system's predictions whether an article is considered fake news or not. In a collaborative effort with other researchers, we develop a machine learning method for document-level stance detection, which labels a document as agreeing, disagreeing, discussing, or being unrelated to a given claim, which is a crucial first step towards automatic fake news detection. In Hanselowski et al. (2018), we detailed the analysis of top-3 systems (including our system) participated at fake news stance detection task. Although there is some overlap of the use of machine learning models, the underlying models are different from the ones in the thesis.

1.5 Thesis Overview

The overview of the organization of thesis and the content of the thesis is as follows:

Chapter 2: Information Preparation: Overview and Background
In Chapter 2, we start by defining information preparation in journalism that motivates this research and structure it into three areas: summarization, condensation and recommendation. Furthermore, we present one use case of each and explain them with an application in practice. To understand the current strand of research in each of the three areas, we give an overview of the existing approaches and discuss them in comparison to the human-in-the-loop based approaches.

Chapter 3: Research Data
In Chapter 3, we describe the data available for the individual areas and also propose a new dataset for a new summarization task in journalism. First, we discuss the live blog summarization task and show the need for such data to develop automatic summarization systems to ease the task of journalists. Later, we propose methods to collect and clean the data to create a benchmark live blog summarization dataset. We end the chapter by analyzing the dataset in terms of size, summarization ratio, heterogeneity, difficulty of the task as compared to the existing popular summarization datasets.

Chapter 4: Information Summarization
In Chapter 4, we introduce the text summarization task as one specific use case of information summarization. First, we motivate the task and discuss the need for human-in-the-loop approaches. We further discuss the limitations of the current state-of-the-art systems by comparing them with the upper bounds on benchmark datasets and the dataset introduced in the previous chapter. To this end, we propose a new interactive summarization methodology with joint optimization techniques to minimize the number of iterations and feedback. Finally, we

introduce our system demonstration which guarantees interactive speeds for large text collections to ensure the user engagement.

Chapter 5: Information Condensation
In Chapter 5, we introduce text compression of new headlines as a use case of information condensation in journalism. First, we motivate the task and discuss the need for human-in-the-loop approaches for annotation and domain adaptation. In the following sections, we propose a novel data-efficient interactive text compression methodology with active learning based sampling strategies for user annotations. We quantitatively evaluate our proposed methodology using state-of-the-art neural sequence-to-sequence text compression models on two popular datasets and analyze the models transfer to new domains with minimal human supervision.

Chapter 6: Information Recommendation
In Chapter 6, we introduce item recommendation as a use case of information recommendation, where an item can be books, products, news articles, etc. To begin with, we motivate the task and discuss the need for models which can both capture user preferences and explain the recommendations. To this end, we propose a multi-task learning of rating prediction and summarization of reviews using attention-based pointer-generated networks. We learn the user and item latent vectors which are shared across rating prediction and review summary generation components. In the later parts of the chapter, we empirically provide evidence for our proposed model being beneficial for recommendation. Finally, to enable the explanations of the learned model we provide (a) user vector visualization on different aspects of the item, (b) a summary of reviews, and (c) the attention highlights on the review based on latent vectors.

Chapter 7: Conclusion
In final Chapter 7, we summarize the findings of the thesis and outline promising directions for future research in information preparation with human-in-the-loop.

Chapter 2

Information Preparation: Overview

In this chapter, we introduce relevant related work on information preparation. First, we define the application scenarios pertaining to information summarization, condensation and recommendation that motivates this research and then illustrate them with several practical examples in journalism. Second, we review the related work in these areas. To this end, we discuss works done with and without the human in the loop.

2.1 Information Preparation in Journalism

In the recent years, digital news and online news consumption has changed significantly across the globe. There is no doubt that the increase of electronic sources is the key contributor to the volume of the information accessed. This trend is likely to continue, as it can be seen from across news organizations. Let us take the example of The New York Times, the 5^{th} most popular news websites across the globe has 300 million unique visitors every day[5] as compared to its 500,000 print edition circulation[6]. Digesting news online offers many benefits over traditional media outlets. The accessibility of news sources geographically across the web and their availability being free of charge makes the web a popular medium. The benefits of online news also yield challenges. There are thousands of news agencies, content providers creating dozens of daily stories and a reader interested in a particular topic is constantly overwhelmed by the amount of information. This negates the benefits of online news as finding relevant stories becomes practically impossible. Furthermore, if we present all the information to the user, they will drown in information.

In the recent literature, Hoq (2016) discuss the primary reasons for the cause of user drowning in information to be the existence of multiple sources with irrelevant and relevant informa-

[5] https://www.similarweb.com/website/nytimes.com#overview last accessed on 11^{th} April 2019

[6] http://www.annualreports.com/HostedData/AnnualReports/PDF/NYSE_NYT_2017.pdf last accessed on 11^{th} April 2019

tion, difficulty in managing information, and lack of user's time to consume the information. Thus, there is a need for information preparation. However, information preparation is hard, and it requires to know the needs of the users, importance is subjective, it has to be fast to enable user engagement, it requires the knowledge of various news domains, e.g., politics, sports, etc., it has to adapt to the user's needs, and many more. Thus, insufficiently prepared information is a significant problem for readers, and we need natural language processing methods that integrate the reader's preferences to solve the problem.

Leveraging natural language processing (NLP) methods for journalism is an emerging research topic. The SciCAR conferences[7] and the recent "Natural Language Processing meets Journalism" workshops (Popescu and Strapparava, 2017, 2018) are predominant examples of this development. Various NLP applications have been used in media newsrooms including traditional printed newspapers, broadcast as well as digital media. These applications can be broadly classified as (a) information preparation for journalists, and (b) information preparation for aiding users with information access.

Information preparation for journalists. The primary activity of the journalist is to produce news stories that present or analyze a topical event. One such application for preparing a published version of a news story is a *proofreading* software. Most of the work in proofreading is proprietary, for example, Grammarly[8], Microsoft Office [9], Writefull[10], etc. A typical proofreading software uses language models trained on millions of articles and provides corrections in terms of vocabulary, punctuation, and grammar. Another use case of NLP application for journalists is the *news headline generation*. A news headline generation is a concise summary of the news article. There have been a variety of previous works in this area to generate the headlines to aid the journalists automatically. Takase et al. (2016) proposes a neural headline generation using Abstract Meaning Representation (AMR). Other popular approaches include summary generation using attention-based neural network models (Rush et al., 2015) and headline generation using generative adversarial networks (Wang and Lee, 2018).

One more application for journalists is *moderating comments and abusive language*. News websites usually allow their readers to comment on news articles, gather feedback, and engage readers. However, user comments can be abusive, i.e., hateful, profane, bullish, which damages the reputation of the news agency and also put off other readers. Various approaches propose to moderate these comments automatically: Kolhatkar and Taboada (2017) propose a multi-classifier approach using Support Vector Machine (SVM) and useful features like argumentation, words, and text quality. Pavlopoulos et al. (2017) uses state of the art recurrent neural network-based moderation model using user type embeddings and user type biases.

[7] https://www.scicar.de
[8] https://grammarly.com
[9] https://www.microsoft.com/en-us/microsoft-365/
[10] https://writefull.com

Lastly, another area of NLP, which is popular, is cross-lingual information retrieval. In this global day and age, very often than not, the information that a journalist requires might not be available in the native language. Steinberger (2013) present a tool called Europe Media Monitor, which helps journalists monitor news in many languages. Another line of research by Rupnik et al. (2016) addresses the problem of tracking of events in a large multilingual stream using cross-lingual document-similarity.

Information preparation for consumers. For news consumers to reduce the problem of information overload, the use of NLP technologies is imminent, and one such application is *news summarization*. News summarization helps the users to consume important information in the shortest time. Radev et al. (2005) propose NewsInEssence, a news summarization system acting as a user's agent to gather and summarize information on news articles. A few other news summarization applications are (a) *multi-document summarization*: summarizing multiple documents on the same news topic or event, (b) *update summarization* (Dang and Owczarzak, 2008): updating the summaries of the news article with the change in contents, (c) *real-time summarization* (Lin et al., 2016): monitor the stream of news content to keep a user up to date on topics of interest, and (d) *live blog summarization* (P.V.S. et al., 2018b): a summary of news article providing coverage of a live event.

Another NLP application to aid people with hearing impairments to understand news telecast is *closed caption generation*. Closed captions are essentially text content that is inserted into a video broadcast. Typically for shows without a live broadcast, captioning is done by a stenographer. However, for live news, telecast text captioning is done using an automatic speech recognition system (Bender and Chesnais, 1988). Additionally, a closely related application is text compression of closed captions. Text compression of closed captions is required as the complete spoken content cannot be displayed on the screen due to the time and space constraints. Luotolahti and Ginter (2015) use SVM and Conditional Random Field (CRF) based sequence classifier using syntactic features.

One more application to reduce the information overload for the news available online is *article recommendation*. Blendle[11] is a New York Times backed startup that built a platform for users to explore news content. Blendle uses online learning to rank (Odijk and Schuth, 2017) setup with a combination of enriched articles and user profiles for article recommendation. Other such news recommender systems are Clavis by Washington Post[12], which uses a hybrid system of both content-based and collaborative filtering to capture both personalization and popularity respectively.

[11] https://blendle.com/

[12] https://knightlab.northwestern.edu/2015/06/03/how-the-washington-posts-clavis-tool-helps-to-make-news-personal/

In this thesis, we are especially interested in information preparation solutions in journalism from the perspective of information access for consumers, and these can be broadly classified as:

- *Information Summarization*: The goal of information summarization is to discover relevant information from multiple sources and prepare data that is relevant for a user's information need. Information summarization in journalism consists of applications like news summarization systems which summarize a news article (or multiple articles), storyline generation of news events, user comments summarization , etc.

- *Information Condensation*: The general definition of information condensation is to reduce the amount of the information needed to represent data. Information condensation can be seen as a filtering methodology to reduce the information overload. In journalism, sentence compression is a major use case which can benefit a wide range of applications, e.g., automatic headline generation, extraction of Twitter news highlights, etc., especially benefit applications on mobile devices which have restricted screen spaces.

- *Information Recommendation*: The goal of information recommendation approaches is to track the user's interests and recommend items based on their preferences. These preferences could be the news content they prefer, the amount of time they spend on a particular article, what opinions they have on a particular topic, etc. The applications of information recommendation applications are not limited to recommending news products (e.g., articles interesting for a user), but also closely linked to news content analysis, such as fake news detection, argumentation mining, or sentiment analysis.

In the following sections, we will discuss each of these scenarios in detail with alleviating information overload. We will specifically discuss one application each related to journalism.

2.2 Information Summarization

Information summarization is the process to gather information from multiple sources and prepare the information for the user. The value added through summarization is provided either manually or automatic services called aggregators. In a broader sense, information providers like newspapers, professional journals are all information aggregators, as they collect information from multiple sources and disseminate the information to be consumed by their audience. Advantages of the information aggregators include the heterogeneity of the data sources, diversity on a topic for a user, supply of multiple sources of information and also foster customization. However, due to the availability of lots of news articles it is still hard to extract useful information. Additionally, it is still a significant problem to acquire information based on user specific needs from the massive collection of news articles.

To alleviate information overload in journalism using information summarization, we first need to understand how a journalist prepares a news story. Traditionally, a journalist is trained to use an inverse pyramid structure (Kovach and Rosenstiel, 2007). The first paragraph of a news story, or the lead, is a concise summary statement of the most important or most interesting, usually addressing who, when, where, why, what, and how of the news article. An article typically begins with an overview of the news story with the most important information, followed by the details of the story. These article structures also correlate with the reading patterns of the users.[13]

A key technique which can exploit news structures to summarize and digest news articles is automatic text summarization. An automatic summarization system can reduce reading time, make the selection process easier and can increase the processing of the texts as compared to the manual ones (Torres-Moreno, 2014). In NLP research, automatic text summarization is the prototypical information summarization task. Given a single or multiple input documents, the goal of text summarization is to create a summary by extracting salient sentences from the input documents. The task of multi-document summarization is challenging as the content and writing styles of the sources vary significantly. In the following subsections, we describe various approaches proposed in the field of text summarization.

2.2.1 Extractive Summarization

Extractive text summarization mainly involves the selection of sentences or phrases from the input documents to put them together to form a summary (Ko and Seo, 2008; Nenkova and McKeown, 2012). The generated summary is a collection of original grammatical elements, which reduces to a combinatorial optimization problem (McDonald, 2007). Extractive summarization can be formally defined in Definition 1.

To solve such combinatorial problems, summarization systems have leveraged powerful techniques like Integer Linear Programming (ILP), submodular maximization, graph-based and ranking-based approaches. In order to score sentences and phrases, Luhn (1958) initially introduced the simple, but influential idea that sentences containing the most important words are most likely to embody the original document. This hypothesis was experimentally supported by Nenkova et al. (2006) who showed that humans tend to use words appearing frequently in the sources to produce their summaries. Many subsequent works exploited and refined this strategy. For instance, by computing TF·IDF (Sparck Jones, 1972) or likelihood ratios (Dunning, 1993).

[13] https://www.nngroup.com/articles/how-users-read-on-the-web/ last accessed on 11^{th} April 2019

Definition 1: Extractive Summarization

Given a set of document D consisting of $\mathcal{S}(D)$ sentences:

- The goal is to select S a subset of sentences from $\mathcal{S}(D)$ such that the length of the summary is less than $\mathcal{L}$ words i.e., $\sum_{s \in S} \text{len}(s) \leq \mathcal{L}$.
- While selecting the subset $S \subset \mathcal{S}(D)$ the system has to make sure that it is the best representative summary of the document collection D.

$$S = \underset{S \subset \mathcal{S}(D)}{\text{argmax}} \sum_{s \in S} \text{score}(s) \quad \text{s.t.} \sum_{s \in S} \text{len}(s) \leq \mathcal{L} \tag{2.1}$$

This is usually formalized as an optimization problem maximizing the score of sentence collection as shown in Equation 2.1, where $\text{score}(s)$ is a scoring function.

Words serve as a proxy to represent the topics discussed in the sources. However, different words with a similar meaning may refer to the same topic and should not be counted separately. This observation gave rise to a set of important techniques based on topic models (Allahyari et al., 2017; Blei et al., 2003). These approaches can be divided into sentence clustering (Radev et al., 2000), Latent Semantic Analysis (Deerwester et al., 1990; Gong and Liu, 2001), and Bayesian topic models (Blei et al., 2003).

Graph-based methods form another powerful class of approaches which combine repetitions at the word and at the sentence level. They were developed to estimate sentence importance based on word and sentence similarities (Mani and Bloedorn, 1997, 1999; Mihalcea and Tarau, 2004). One of the most prominent examples is LexRank (Erkan and Radev, 2004).

More generally, many indicators for sentence importance were proposed and therefore the idea of combining them to develop stronger indicators emerged (Aone et al., 1995). Kupiec et al. (1995) suggested that statistical analysis of summarization corpora would reveal the best combination of features. For example, the frequency computation of words or n-grams can be replaced with learned weights (Hong and Nenkova, 2014; Li et al., 2013). Additionally, structured output learning permits to score smaller units while providing supervision at the summary level (Li et al., 2009; Peyrard and Eckle-Kohler, 2017).

A variety of works proposed to learn importance scores for sentences (Yin and Pei, 2015; Cao et al., 2015a). This started a large body of research comparing different learning algorithms, features and training data (Hakkani-Tur and Tur, 2007; Hovy and Lin, 1999; Wong et al., 2008a). The feature-based approaches largely depend on feature engineering to determine the performance of the summarization system. The proposed feature based approaches fall into two categories: document-dependent (Ren et al., 2016) and document-independent or context-free (Hong and Nenkova, 2014; Cao et al., 2015b; Wan and Zhang, 2014) approaches.

Although many document-dependent features for sentence ranking have been proposed, the document-independent feature-based approaches have outperformed them and adapt better across datasets (Cao et al., 2015b). Ren et al. (2016) propose a redundancy-aware sentence ranking approach using two neural networks with handcrafted features. Hong and Nenkova (2014) propose a sentence ranking approach based on handcrafted document-independent features as the summary priors. On the other hand, Cao et al. (2015b) propose a novel summarization system PriorSum, which uses an enhanced convolution neural network to capture the context-free summary prior features. The latest work by Ren et al. (2018) propose an approach which uses the encoded sentence representation of PriorSum with sentence relations to improve the performance in sentence regression.

Nowadays, due to the recent success of neural networks and availability of large-scale training data in various NLP tasks, sequence-to-sequence methods are employed (Nallapati et al., 2017; Kedzie et al., 2018; Yasunaga et al., 2017; Narayan et al., 2017) to extractive summarization. These methods rely on recurrent neural networks (RNNs) to derive input document representation which is then used to label sentences to be part of a summary. Nallapati et al. (2017) propose SummaRunner, which is a two-layer RNN with a set of hand-crafted features. In another strand of research where there is lack of sufficient training data and diverse categories of documents, Cao et al. (2017) propose the use of text classification datasets with convolutional neural networks to learn better document representations which explores summary styles with respect to the text categories.

2.2.2 Abstractive summarization

Extractive summarization approaches have several problems: (1) inclusion of unimportant details due to complete sentence collection, in the process of selecting important sentences it might include sentences having unimportant details with the important ones, (2) extractive summaries are non-cohesive and lack fluency, as the selected sentences might contain dangling and unresolved pronouns (Nenkova and McKeown, 2012). In contrast to extractive summarization, abstractive summarization aims to produce new and original texts (Khan et al., 2016) either from scratch (Rush et al., 2015; Chopra et al., 2016), by fusion of extracted parts (Barzilay and McKeown, 2005; Filippova, 2010), or by combining and compressing sentences from the input documents (Knight and Marcu, 2000; Radev et al., 2002). Intuitively, abstractive systems have more degrees of freedom. Indeed, careful word choices, reformulation and generalization should allow condensing more information in the final summary. An abstractive summarization approach can be formalized as in Definition 2.

Definition 2: Abstractive Summarization

Given an input document x consisting of a sequence of N words $x_1, x_2, ..., x_N$ comprising of a fixed vocabulary V of size $|V|$:

- The goal is to output a shortened text $y = y_1, y_2, ..y_M$ of length $M < N$. Unlike related tasks like machine translation, the output length of the summary M is known to the system and is fixed before generation.
- The goal of the system is to find an optimal sequence of summary y from Y as in Equation 2.3:

$$\text{argmax}_{y \in Y} \text{ score\_pair}(x, y) \tag{2.2}$$

where Y is a set of all possible sentences of length M and $\text{score\_pair} : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ is a scoring function. The scoring function is typically modelled as a local conditional distribution and this function varies for different architectures.

Recently, end-to-end training based on the encoder-decoder framework with long short-term memory (LSTM) has achieved huge success in sequence transduction tasks like machine translation (Sutskever et al., 2014). For abstractive summarization, large single-document summarization datasets rendered possible the application of such techniques. For instance, Rush et al. (2015) introduced a sequence-to-sequence model for sentence simplification. Later, Chopra et al. (2016) and Nallapati et al. (2016) extend this work with attention mechanisms. The models ability to produce fluent summaries thereby substantially increased the interest in abstractive summarization. Additionally, since words from the summary are often retained from the original source, copy mechanisms (Gu et al., 2016; Gulcehre et al., 2016) and strategies to avoid repetitions in the generated summary have been thoroughly investigated (See et al., 2017). The previous works of attention-based encoder-decoder models have been later modified by various models such as a guided generation model (Li et al., 2018), a two step summarization model which first selects the important sentences and rewrites them abstractively (Chen and Bansal, 2018), a hierarchical structured self-attention mechanism to incorporate the knowledge of the document structure by creating the sentence and document embeddings (Al-Sabahi et al., 2018) and a generative adversarial network Liu et al. (2018a).

Although abstractive summarization shows promising results in generating text, these results are not yet state-of-the-art as compared to extractive approaches for multi-document summarization, as it is hard to generate readable, complete and grammatical sentences. However, impressive results have been achieved on constrained generation settings such as a generating headlines (Rush et al., 2015) or generating summaries of a news article (See et al.,

2017) or a review (Li et al., 2017b). The advantage of these approaches is that the approaches are completely data-driven and can be trained end-to-end.

2.2.3 Human-in-the-loop Summarization

One of the major problems of automatic extractive and abstractive summarization systems is the lack of producing high quality summaries on par with human summaries. Human-in-the-loop based summarization techniques have been proposed to help users deal with creating personalized high-quality summaries. Attempts have been made to replicate human understanding of summary production in restricted domains (DeJong, 1982), however, the vast amount of knowledge required by these systems makes it impractical to port to other domains.

Kay (1997) first proposed machine-aided translation approach defined as "a cooperative man-machine translation system". This inspired the work in the human-in-the-loop summarization, whose essential goal is to help humans in the summarization process. In this way, human effort and time is reduced in making the summary. Endres-Niggemeyer (1998) first confirmed the feasibility of the human-in-the-loop summarization approaches by first identifying three stages of human summarization: (1) *document exploration*, (2) *relevance assessment*, and (3) *summary production*. Overall main topics of the text and the structure are identified in the first two stages, followed by editing the copied text in the third stage.

Endres-Niggemeyer (1998) develop a computer-aided summarization tool which automatically identified the most important sentences in the text. Creating a summary then requires the human to cut, paste, and reorganize the important elements in order to formulate a final text. Following this research, some other human-in-the-loop summarization tools have been proposed (Craven, 2000; Narita et al., 2002; Orăsan et al., 2003; Orăsan and Hasler, 2006). Craven (2000) propose a computer assisted summarization system where users were presented with automatically extracted phrases and they were asked to reformulate abstracts. Narita et al. (2002) propose a web-based abstract writing tool for helping Japanese software engineers improve their content organization for writing by enabling them to select an abstract template and sample sentence constructions. Orăsan et al. (2003) propose a computer-aided summarization system which combines several summarization methods, where the users are allowed to interact with parameters of the system and output to improve the summary quality.

While most previous work focuses on generic summaries, there have been a few attempts to take a user's preferences into account. The study by Berkovsky et al. (2008) shows that users prefer personalized summaries that precisely reflect their interests. These interests are typically modeled with the help of a query (Park and An, 2010) or keyword annotations reflecting the user's opinions (Zhang et al., 2003).

Another summarization task which has human in the loop is the real-time summarization task. This task began at the TREC 2016 and represents an amalgam of the microblog track and the temporal summarization track (Lin et al., 2016). In real-time summarization, the goal is to

automatically monitor the stream of documents to keep a user up to date on topics of interest and create email digests that summarize the events of that day for their interest profile. The drawback of this task is that they have a predefined time frame for evaluation due to the real-time constraint, which makes the development of systems and replicating results arduous.

2.3 Information Condensation

The concept of information condensation comes from the notion of data compression in information theory, where the data is compressed by encoding information into fewer bits than the original representation (Shannon, 1948). Information condensation is a similar process defined as the process to reduce the information and prepare a concise version of it to the user. In journalism, the value added through information condensation is primarily used in the editorial process. The role of an editor is to condense the vast amount of material provided by the reporters such that it fits the the size constraint of the paper. One way to condense information in the text is by compressing information in a text called text compression.

Text compression is the natural language generation (NLG) task of condensing a sentence while preserving its most important contents. It has many real world applications, such as compressing micro-blog, generating headlines of the newswire article. Beyond journalistic use cases, text compression has a wide variety of useful applications where there is space constraint like subtitle generation for speech transcripts to be displayed on screen in parallel with the audio and video content (Vandegehinste and Pan, 2004), audio scanning devices for blind (Grefenstette, 1998).

Text compression approaches can be broadly classified into two: (a) deletion-based, and (b) abstractive .

2.3.1 Deletion-based Text Compression

Jing (2000) introduce a definition of text compression as "to reduce without major loss", which formulates text compression as removing as many extraneous words as possible from the text without diminishing the main content of the text. In this definition the goal is to delete unimportant words from a source text to generate a shorter version of the text without any sentence transformations. Much of the research in text compression literature has followed this simplified approach of removal of words from the original text. Marsi et al. (2010) characterize the task in terms of two assumptions: (1) only word deletions are allowed and (2) the word order cannot be altered. Examples of these transformations can be seen in the examples 1. and 2. below, where (i) is the original text and (ii) is the corresponding deletion-based compression.

For example:

1. (i) German Chancellor Angela Merkel has said she will fight for an "orderly Brexit" until "the very last hour".
   (ii) *Angela Merkel will fight for an orderly Brexit.*

2. (i) French President Emmanuel Macron says France would 'probably' have voted to leave the EU, if offered the choice in a referendum.
   (ii) *Emmanuel Macron says France would probably have voted to leave the EU.*

The deletion-based text compression can be formalized as in Definition 3:

Definition 3: Deletion-based text compression

Given an input sentence x consisting of a sequence of N words $x_1, x_2, ..., x_N$:

- The goal is to output compressed text $y = y_1, y_2, .., y_N$ where $y_i \in \{0, 1\}$ by deciding whether to keep or drop the input token (Knight and Marcu, 2002).
- But retain the most important information and remain grammatical.

Much of the work on text compression in literature is on deletion-based approaches. The approaches are split into approaches which are data intensive and the other which are data lean. Due to minimal training data available earlier research primarily focused on data lean approach. The data lean approaches followed an unsupervised paradigm and learn grammaticality from large amounts of text. Clarke and Lapata (2007) propose an integer linear programming based approach, which incorporates surrounding discourse information instead of compressing sentences in isolation. To this end, rather than using training pairs of sentence–compression pairs, they use a large language model to find the most probable compressed sentence.

The data intensive approaches followed a supervised learning approach and required parallel data of sentence–compression pairs. These methods explored various modeling approaches, including the noisy-channel model (Knight and Marcu, 2002; Turner and Charniak, 2005), variational autoencoders (Miao and Blunsom, 2016), and Seq2Seq models (Filippova et al., 2015). Knight and Marcu (2002) introduce noisy-channel model approaches which are based on syntactic tree structures i.e., produced a compressed syntactic tree. Extending their work, Galley and McKeown (2007) propose lexicalized markov grammars to estimate the probabilities of these syntactic trees and Turner and Charniak (2005) propose an improved model by replacing the ad-hoc language model with syntax-based language model (Charniak, 2001). Instead of using the tree-based transformations, McDonald (2006) propose sentence-based approaches by finding the highest scoring sentence directly. They used features from constituency and dependency trees on compressed bigrams and used a discriminative large-margin learning framework. However, most of these works are dependent on parsing systems and this makes the sys-

tems vulnerable to parsing error propagation. To address this, Filippova et al. (2015) propose a robust compression model which benefits from the advances in deep learning methodologies like Long Short Term Memory models (LSTMs). This system produced surprisingly readable and informative compressions. The RNN based models typically required to be trained on large data sets of aligned sentence–compression pairs. To this end, Filippova and Altun (2013) propose creation of large training data for deletion-based text compression created from news headlines. Following this research, Wang et al. (2017c) propose the use of syntactic features by adding as input features to LSTM model and as hard constraints on the compressed text. Zhao et al. (2018) propose a language-model based evaluator which is a syntactic neural language model. A series of trial-and-error deletion operations are performed on the source text and then a reinforcement learning framework is used to obtain the best target compression.

2.3.2 Abstractive Text Compression

In contrast to deletion-based approaches, abstractive models generate a shorter text by inserting, reordering, reformulating, or deleting words of the source text. Examples of abstractive text compression can be seen in the examples 1. and 2. below, where (i) is the original text and (ii) is the corresponding abstractive compression.

For example:

1. (i) German Chancellor Angela Merkel has said she will fight for an "orderly Brexit" until "the very last hour".
   (ii) *Angela Merkel strongly supports orderly Brexit.*

2. (i) French President Emmanuel Macron says France would 'probably' have voted to leave the EU, if offered the choice in a referendum.
   (ii) *Emmanuel Macron says if offered a choice, France would have voted to leave the EU.*

The abstractive text compression can be formalized as in Definition 4:

Definition 4: Abstractive Text Compression

Given an input sentence x consisting of a sequence of N words $x_1, x_2, ..., x_N$:

- The goal is to output a shortened text $y = y_1, y_2, ..y_M$ of length M where M/N is the compression ratio.
- Find an optimal compression sequence y from Y as in Equation 2.3:

$$\text{argmax}_{y \in Y} \text{ score\_compression}(x, y) \tag{2.3}$$

where Y is a set of all possible sentences of length M and $\text{score\_compression} : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ is a scoring function. The scoring function is typically modelled as a local conditional distribution.

The generative approaches have received a lot of attention recently. Given a source sentence x and target compression y, these models typically estimate the joint probability $P(x, y)$. The initial use of these models is inspired from machine translation as they are similar tasks. In translation, the goal is to translate source language text into a target language text, whereas in sentence compression instead of translating between two languages we are translating between source text and target compression. Abstractive text compression can also be seen as a "scaled down version of the abstractive text summarization problem" (Knight and Marcu, 2002), as the task is on the level of sentences instead of complete documents (see Definition 2 in Section 2.2.2).

Recent abstractive models have seen tree-to-tree transduction models proposed by Cohn and Lapata (2009). The sentence compression is formulated as a tree-to-tree rewriting task. The model uses synchronous tree-adjoining grammars (Shieber and Schabes, 1990) to capture all possible rewrites for a sentence and also structural mismatches. The grammar rules are assigned weights which are learned discriminatively using a large margin technique (Tsochantaridis et al., 2005). Cohn and Lapata (2013) follow up on their previous work and describe a model that can handle mismatches on the structural and lexical level.

Recently, significant advances have been made in Seq2Seq models in machine translation (Sutskever et al., 2014) and abstractive summarization (Chopra et al., 2016) has also encouraged variations of these for abstractive text compression. The modeling has been shifted from traditional approaches of feature engineering to more focused parameter optimization models that learns mappings between sequences by learning end-to-end representations. These approaches use RNNs to encode the source text into a fixed vector with fixed length and decode into the target compression. Rush et al. (2015) propose an attention-based RNN for abstractive headline generation inspired from machine translation (Bahdanau et al., 2015). Wubben et al.

(2016) propose attentive LSTM models for caption or scene description compression. Lastly, work by Yu et al. (2018) propose operation networks where the Seq2Seq model decoder is replaced with a deletion decoder and a copy-generate decoder.

In another strand of research in the field of sentence simplification, there have been some more work to simplify text similar to text compression (Coster and Kauchak, 2011; Woodsend and Lapata, 2011; Narayan and Gardent, 2014). Coster and Kauchak (2011) develop sentence simplification models from Simple English Wikipedia articles paired with its corresponding English Wikipedia articles. Their models performed reordering, insertion and paraphrasing actions in addition to deletion. Woodsend and Lapata (2011) model a quasi-synchronous grammar and integer linear programming method using the edit histories of Wikipedia and Simple Wikipedia. Narayan and Gardent (2014) propose a hybrid approach which combines deep semantics and monolingual machine translation to derive simple sentences.

2.3.3 Human-in-the-loop Text Compression

Given a text, to compress it requires the knowledge of important information. However, importance of information in a source text is very subjective and it will depend on various factors dependent on the user like their background knowledge, information need. For example, in the example presented above if the user is aware of the background knowledge that Angela Merkel is the Chancellor of Germany and Emmanuel Macron the President of France, then the respective compressions are appropriate without any loss of information. Furthermore, background knowledge can also be gathered while reading the sentences which typically contain redundant information. Information need is another factor which also influences the compression. User's information need could be specific to the task at hand. For example, one the one hand, compressing to generate microblogs while covering live events in the case of live blogs, requires the user to generate eye-catching headlines to keep audience updated. On the other hand, compressions for a generic domain would be different. The information need of a user can be gathered based on the user feedback. Let us consider the case of a model learned on large compression dataset of sentence–headline pair and now we want to adapt the model to a new generation dataset. We can let a user compress a few samples and learn to adapt to the user's needs.

To the best of our knowledge, there is little to no work with human-in-the-loop based methods for text compression. Cohn and Lapata (2013) make 15 volunteers to use an interactive system to examine human compression, however, they do not learn from the user feedback. Toutanova et al. (2016) create a new dataset for sentence compression, where they used crowd-source platform to compress text representing business, newswire, journals, and technical documents. The authors recorded and analyzed the edit history of the user rewrite operations but did not use it to enhance the model.

There is some work in the area of machine translation which uses similar problem setting and models to text compression, the only difference being generation of foreign language text instead of text from the source. Interactive machine translation intention was to let human translators and the machine translation system to work in tandem. Nepveu et al. (2004) propose an interactive system with dynamic adaptation using cache for language models and translation models. CASAMACAT by Alabau et al. (2014) is one such tool which presents the user with the target output and the user post-edits it submits the corrected output. The system learns from the user corrections and reduces the human effort in the long run. Pérez-Ortiz et al. (2014) propose an interactive machine translation with a resource-agnostic approach, where the suggestions are obtained from any bilingual source. González-Rubio et al. (2012) and Peris and Casacuberta (2018) propose active learning for interactive machine translation to efficiently sample data for translating data streams.

2.4 Information Recommendation

Information recommendation is defined as a process to predict whether a user will like an item or present the users with a set of items which might be of interest to them. With the increase of content on the WWW, information recommendation systems have become an important building block of many online web applications. Information recommendation in journalism is usually carried out by News Media giants who have access to a broad range of user information which they have gathered from user interactions from their websites. The goal of the news agencies is to reduce the clutter of news from all over world and present it to their audience. Therefore, information recommendation algorithms are developed by many news websites and news content is recommended and personalized for the users while utilizing user's specific preferences. These preferences include topics in which a user is interested in, reading and click patterns of a user.

Definition 5: Information Recommendation

Given we have $\mathcal{U}$ users and $\mathcal{V}$ items, and $\mathcal{R}$ denotes the user–item interaction matrix

- The goal of a recommender system is to predict the user–item interaction matrix $\hat{\mathcal{R}}$ which comes close to the ground-truth $\mathcal{R}$.

An alternative formulation of the task is:
Given a user u_i,

- present a list of items $\{v_1, v_2 ...\}$ that the user likes based on $\hat{\mathcal{R}}$.

Information recommendation systems are broadly classified into : (a) collaborative filtering, and (b) content-based filtering discussed below.

2.4.1 Collaborative Filtering Recommender

In collaborative filtering, items are recommended to a user based on collecting preferences from many users, thus the word collaborative. Collaborative filtering has its roots in information filtering and information retrieval. This approach assumes that if a user u_1 has the same interest as user u_2 in an item, then u_1 is more likely to have the same interest in a new item as u_2 than that of any randomly chosen user. For example, a collaborative filtering recommender system for a news website would recommend news articles to a user not only based on the user's interests of likes and dislikes, but also based on information gathered from many users.

Goldberg et al. (1992) proposed Tapestry the first collaborative filtering recommender system, which was an electronic messaging system that allowed users to rate messages. The name has since been referred to any system relying on other users' interests for contents like information on restaurants, movies, shopping, books, research articles, etc. Collaborative filtering approaches typically analyze the relationships between users and items in a domain and identifies user–item associations Hu et al. (2008).

Collaborative filtering methods have been successful for a long time in recommendation systems (Deshpande and Karypis, 2004; Marlin, 2003; Koren, 2008; Salakhutdinov and Mnih, 2007; Lee and Seung, 2000; Koren, 2008). Salakhutdinov and Mnih (2007) proposed probabilistic matrix factorization (PMF), which is a Matrix Factorization method using Gaussian distribution to model the users and items latent factors. This approach scales linearly with the number of user–item observations. They further extend the PMF model to include an adaptive prior on model parameters. Another variation of the PMF is the non-negative matrix factorization (NMF) proposed by Lee and Seung (2000), which factorizes the rating matrix into a user matrix and item matrix to have no negative elements.

One of the popular collaborative approach which won the Netflix prize[14] is singular value decomposition (SVD). SVD is a matrix factorization technique which reduces the number of features of a data set by reducing the latent space dimensions. As an extension, Koren (2008) proposes SVD++ by moditifying the prediction to also include the effect of the implicit information as opposed to only explicit information in the previous model. SVD++ leverages the strengths of both the neighbourhood model as well as the latent model.

In the recent years, the rise of deep learning techniques also contributed to advances in collaborative filtering methods. Neural architectures provided end-to-end differential frameworks where the models are able to exploit the inherent structure in the data. Some of these models are Neural Collaborative Filtering (NCF) (He et al., 2017, 2018), Factorization Machines (He and Chua, 2017), Deep Matrix Factorization (Xue et al., 2017). He et al. (2017) present a

[14]https://www.netflixprize.com/

NCF framework to learn non-linear interactions between users and items. Later, He and Chua (2017) proposed the Neural Factorization Machines by modeling higher-order and non-linear interactions. Zhou et al. (2016) propose a social factorization machines, which combines the social information to the neural collaborative filtering methods.

Collaborative filtering models have been successfully applied by representing users and items in a shared, low-dimensional space. Vectors in this space represent latent factors of users and items. Using the dot product of two vectors, one can predict a user's rating for a given item. The drawback of these approaches is that the performance of the systems degrades when the rating matrix is sparse, the so-called cold-start problem (Esparza et al., 2011). This setting is often observed for systems developed on domains with small data. Also, their effectiveness is limited when the users have difficulty in expressing their preferences as scalar ratings on items (Wing-ki Leung et al., 2006). Another drawback of these approaches is that the user and item vectors of the latent space cannot be interpretable, which hampers providing an explanation that can be understood by the users. This motivated researchers towards content-based filtering models for recommendation.

2.4.2 Content-Based Filtering Recommender

In contrast to collaborative filtering, content-based filtering recommender systems builds the user and item profiles independent of the preferences of other users. These preferences are extracted from content representations of items that have similar content to the items liked by the user (Lops et al., 2011). For example, an Apple iPhone X and Samsung galaxy S9 phone have similar properties like camera, display, screen size. If a user is interested in camera and display of an iPhone X he/she would be interested in the same properties for any other phone. Other user-generated information such as tags (Marinho et al., 2011), social network (Chen and Wang, 2014) are also used for better representations for recommendations.

In this work, we particularly emphasize the use of *user reviews* for learning user and item representations. The popularity of social media, e-commerce sites and news media has encouraged users to write reviews or comments describing their opinion of the items. These reviews are typically textual comments that explain their likes and dislikes of an item. The users opinions are naturally multi-faceted and hence can capture fine-grained user preferences. These methods typically learn user (McAuley and Leskovec, 2013) and item profiles (Aciar et al., 2007) from item user reviews. They recommend an item to a user by matching the item's features with that of the user preferences. There are works which identify the importance of aspects for the users by integrating topic models to generate the users' and items' latent factors from review text (Musat et al., 2013).

Content-based filtering using user reviews can deal with the sparsity problem by providing additional information about user preferences. These systems can also help tackle the cold-start problem for new users having limited experience with the items. By aligning the review

information a preference model is constructed for a user with few ratings (Seroussi et al., 2011). Other works include the usage of review helpfulness votes (Raghavan et al., 2012) and review emotions like opinion words (Zhang et al., 2010).

With the growing popularity of deep learning methods to be able to better represent textual content, many content-based recommender systems have popped up in the recent years. The rise of these approaches are because of the ability of deep neural networks to learn underlying explanatory factors and useful representations from input data. Wang et al. (2015) propose Collaborative Deep Learning framework, which jointly performs deep representation learning for the content and collaborative filtering using a hierarchical Bayesian model. Zheng et al. (2017) propose Deep Cooperative Neural Networks, which jointly model users and items from textual reviews using two parallel neural networks coupled using a shared output layer. Chen et al. (2018) propose a similar two neural network architectures as earlier, additionally, they use an attention-based review pooling to select reviews as explanations.

Content-based filtering has an upper hand in solving these problems, however, less research is done in the area of explaining these recommendations. Some recent research showed that recommendations systems can benefit by keeping the human in the loop in the process of making predictions, which we discuss in the next section.

2.4.3 Human-in-the-loop Recommender

Although collaborative filtering and content-based filtering methods recommend items based on other users' preferences or content-based user preferences, the recommendation system needs to take into account user's needs. One such need is that the recommender system to be explainable. Explainable recommendation systems help to keep the human in the loop in the recommendation process, which improves the effectiveness, efficiency and user satisfaction of the recommender systems.

Explainable recommendation was formally introduced by Zhang et al. (2014). The authors propose explicit factor models based on phrase-level sentiment analysis on reviews and present word clouds of aspect–opinion pairs as explanations. These sentiment-based approaches have also been leveraged in social recommendation (Ren et al., 2017) and point-of-interest recommendation (Zhao et al., 2015). Li et al. (2017b) propose a multi-task learning setup by leveraging gated recurrent units to summarize the reviews of an item to generate tips as an explanation. In the recent years, much focus has been on the need for explanation based systems and it is still a budding field.

In another strand of research, Díaz and Gervás (2007) create user models based on social tagging and Hu et al. (2012) rank sentences by combining informativeness scores with a user's interests based on fuzzy clustering of social tags. Extending the use of social content, another recent work showed how personalized review summaries (Poussevin et al., 2015) can be useful in recommender systems beyond rating predictions.

There is a research gap in explainable recommendation while having human in the loop. Efficient methods are required to provide quality and novel information, feeding out relevant information while dealing with problems such as 'data-sparsity' commonly associated with recommending content.

2.5 Chapter Summary

In this chapter, we introduced and defined information preparation in journalism into three areas: summarization, condensation and recommendation as a way to alleviate information overload. A variety of computational models have been developed for each of the tasks mentioned in these scenarios.

In the area of information summarization, we reviewed work on text summarization. Text summarization provides a concise summary of the document collection by aggregating information. These summaries can be used in journalism by journalists to provide summarized information about an article or live event. These summaries are are also useful for readers facing information overload. We reviewed existing text summarization approaches that aim to automate summarizing a collection of documents using extractive and abstractive approaches. These works are based on unsupervised and supervised techniques using no training data to large training data. In particular, we analyzed existing research with the human in the loop which reduces the users effort to process the content and produce personalized summaries. We pointed out the need for such techniques to take advantage of both the worlds by integrating user feedback to collect important information.

Similarly, in the area of information condensation, we particularly discussed the task of text compression. Text compression is a natural second step to information summarization models and they provide a compression of the sentence by removing unimportant information. This task is usually approached in two ways: (1) deletion-based, or (2) abstractive. The goal of deletion-based text compression is to drop unimportant words in the text without reordering or transforming the text. Abstractive text compression is similar to abstractive summarization methods where the goal is to generate text given a source text. Existing work is spread across communities from text simplification, machine translation and abstractive summarization, however, it is still unclear how to efficiently use human in the loop to perform text compression.

Lastly, we discussed the area of information recommendation. The task of item recommendation is popular in scenarios of recommending news articles to the readers, shopping items to customers, scientific research articles to researchers. Recommender systems typically suggest a list of recommendations using one of the two ways (a) collaborative filtering or (b) content-based filtering. Each system has their strength and weakness, however, only a few models have shown how to use explainable recommendation to keep human in the loop while producing the list of recommendations.

Chapter 3

Research Data

In this chapter, we will look into the data available for the three information preparation scenarios i.e. summarization, condensation and recommendation. Additionally, in Section 3.2, we discuss the use case of live blog summarization for information preparation in a journalistic use case, followed by, our novel pipeline to collect and extract the human-written summaries and postings from online live blogs. Section 3.2.4 provides a detailed analysis of the corpus we created from live blogs of two major news publishers, the *BBC* and *The Guardian*, using our pipeline. To conclude, we provide a chapter summary in Section 3.3.

3.1 Existing Research Corpora

In this section, we discuss previous work on three areas of information preparation namely summarization, condensation and recommendation. As introduced in Chapter 2, we describe the existing research data in text summarization, text compression and item recommendation.

3.1.1 Information Summarization

The most widely used summarization corpora have been published in the Document Understanding Conference[15] (DUC) and Text Analysis Conference[16] series. In total, there are human-written reference summaries for a variety of multi-document summarization tasks, such as, generic (DUC'01-'04), query-focused (DUC'05-'06, TAC'08), update (DUC'07-'08) and guided (TAC'09-'11) summarization. Although the research community has often used these corpora, their limited size prevents training advanced methods, such as encoder–decoder architectures, and it is time-consuming and labor-intensive to extend such corpora with large numbers of manually written summaries.

[15] https://duc.nist.gov/
[16] https://tac.nist.gov/

Dataset	Lang	Topics	Domain	# Docs	Summary	
					type	# words
ACL Anthology (Bird et al., 2008)	en	10,921	sci	1	a	100-200
CNN/DailyMail (Hermann et al., 2015)	en	312,084	news	1	a	≈ 50
DUC'01 (Over and Yen, 2001)	en	30	news	10	a	100–400
DUC'02 (Over and Liggett, 2002)	en	59	news	10	a/e	100–400
DUC'03 (Over and Yen, 2003)	en	59	news	30	a	100
DUC'04 (Over and Yen, 2004)	en	50	news	10	a	100
DUC'05 (Dang, 2005)	en	50	news	32	a	250
DUC'06 (Dang, 2006)	en	50	news	25	a	250
DUC'07 (Over et al., 2007)	en	25	news	10	a	100
Opinosis (Ganesan et al., 2010)	en	51	rev	100	a	25
TAC'08 (Dang and Owczarzak, 2008)	en	48	news	20	a	100
TAC'09 (Dang and Owczarzak, 2009)	en	44	news	20	a	100
TAC'10 (Dang and Owczarzak, 2010)	en	46	news	20	a	100
TAC'11 (Dang and Owczarzak, 2011)	en	46	news	20	a	100
TGSum (Cao et al., 2016)	en	204	news	≈ 6	a	≈ 100
MultiLing'11 (Giannakopoulos et al., 2011)	7	10	news	10	a	≈ 250
MultiLing'13 (Giannakopoulos, 2013)	7	10	news	10	a	≈ 250
MultiLing'15 (Giannakopoulos et al., 2015)	10	15	news	10	a	≈ 250
de Loupy et al. (2010)	fr	20	news	20	a	≈ 200
Goldstein et al. (2000)	en	25	news	10	e	≈ 200
Ulrich et al. (2008)	en	30	email	11	a/e	250
AMI (Carletta et al., 2006)	en	137	mtg	1	a/e	300
Zechner (2002)	en	23	speech	N/A	e	$10\% * \|D\|$
Carenini et al. (2007)	en	20	email	≈ 4	e	$30\% * \|D\|$
Nakano et al. (2010)	en	24	hetero	352	e	≈ 350
Lloret and Palomar (2013)	en	310	hetero	10	e	100–200
DBSv1 (Benikova et al., 2016)	de	10	hetero	4–14	e	≈ 500
DBSv2	de	30	hetero	4–14	e	≈ 500
hMDS (Zopf et al., 2016)	en	91	hetero	13	a	≈ 250
auto-hMDS (Zopf, 2018a)	en,de	7,316	hetero	≈ 9	a	≈ 300
Tauchmann et al. (2018)	en	10	hetero	≈ 80	e	N/A

Table 3.1: Overview of the existing datasets for summarization. Abbreviations and Symbols: a: abstract, e: extract, sci: scientific, mtg: meetings, rev: reviews, hetero: heterogeneous, $|D|$: input document size

Large datasets exist particularly for single-document summarization tasks, including the ACL Anthology Reference Corpus (Bird et al., 2008) and the CNN/Daily Mail dataset (Hermann et al., 2015). The latter contains large pairs of 312k online news articles and multi-sentence summaries used for neural summarization approaches (Nallapati et al., 2016; See et al., 2017). Another recent work uses social media posts on Twitter to create large-scale multi-document summaries for news: Cao et al. (2016) use hashtags to cluster the tweets on the same topic, and

they assume the tweet's content to be a reference summary for the document linked by the tweet. Their TGSum corpus consists of 204 document clusters with 1,114 documents and 4,658 reference tweets. Lloret and Palomar (2013) create a similar corpus of English and Spanish news documents and corresponding tweets linking to them.

In contrast to the abstractive summarization corpora, there are corpora which are more suitable for evaluating extractive MDS systems containing only extracts. Goldstein et al. (2000) create a multi-document summarization corpus of 25 sets of 10 newswire articles taken from Yahoo categories, where three annotators create a summary consisting of 10 most informative sentences. Zechner (2002) follow a similar approach to create an extractive summary for spoken dialog summarization. Other extractive summarization corpora include email (Carenini et al., 2007) and heterogeneous (Nakano et al., 2010; Lloret and Palomar, 2013) data. Furthermore, Ulrich et al. (2008) and Carletta et al. (2006) create two different email summarization corpora consisting of abstractive and extractive summaries. They create corpus by asking annotators to first select most important sentences and then write a summary of the email thread. Such an approach creates links between the selected sentences and the human written sentences, which can be used for both extractive and abstractive summarization.

Other multi-document summarization datasets focus on heterogeneous sources: Zopf et al. (2016) and Zopf (2018a) use Wikipedia articles as reference summaries and automatically search for potential source documents on the web. Benikova et al. (2016) propose an expert-based annotation setup for creating a summarization corpus for highly heterogeneous text genres from the educational domain (DBSv1 and DBSv2).[17] In similar lines of research, Tauchmann et al. (2018) use a combination of crowdsourcing and expert annotation to create a hierarchical summaries for a heterogeneous web crawl. In another strand of research, Giannakopoulos et al. (2011) introduce multilingual summarization corpora (Giannakopoulos, 2013; Giannakopoulos et al., 2015), Ganesan et al. (2010) introduce Opinosis a corpus on opinions, and Li et al. (2017a) introduce a corpus of reader-aware multi-document summaries, which jointly aggregate news documents and reader comments.

3.1.2 Information Condensation

Early publicly available text compression datasets are manually curated but small (Knight and Marcu, 2002; Cohn and Lapata, 2008; Clarke and Lapata, 2006, 2008). Ziff-Davis Corpus is one of the earliest sentence compression corpus created by Knight and Marcu (2002) from news articles on computer products. The corpus was constructed by automatically matching the sentence in the article with sentences occurring in the corresponding abstract. Clarke and Lapata (2006) show that Ziff-Davis corpus differs substantially from the manually-created compressions and conclude that this corpus is not suitable for studying compression. Clarke and Lapata (2006) create a manually-crafted sentence-compression corpus by asking three anno-

[17] https://github.com/AIPHES/DBS

Dataset	Lang	Pairs	Domain	Train	Dev	Test
Ziff-Davis (Knight and Marcu, 2002)	en	1,067	news	1,035	-	32
Clarke and Lapata (2006)	en	1,370	news	-	-	-
Clarke and Lapata (2008)	en	1,433	news	-	-	-
Cohn and Lapata (2008)	en	575	news	-	-	-
Google News (Filippova and Altun, 2013)	en	250k	news	195,000	5,000	10,000
Filippova et al. (2015)	en	2M	news	2M	5,000	10,000
Gigaword (Rush et al., 2015)	en	$\approx 4.5M$	news	$\approx 3.8M$	394,622	381,197
MSR OANC (Toutanova et al., 2016)	en	6,169	hetero	4,936	785	448

Table 3.2: Overview of the existing datasets for text compression. Abbreviations and Symbols: hetero: heterogeneous, lang: language

tators to compress sentences by removing tokens. The sentences are from 50 broadcast news stories taken from HUB-4 1996 English Broadcast News corpus provided by the Linguistic Data Consortium. This Broadcast News corpus consists of news from (a) written news corpus from The LA Times, Washington Post, Independent, The Guardian and Daily Telegraph and (b) spoken news corpus consisting of broadcast news from a variety of networks such as CNN, ABC, NPR and CSPAN. The compression corpus consists of 1,370 sentence–compressions pairs.[18] Clarke and Lapata (2008) enlarge the previous corpus following similar manual procedure to construct a corpus with 1,433 sentences of 82 news articles from the British National Corpus (BNC) and American News Text Corpus.[19] Cohn and Lapata (2008) use a subset of this corpus of 575 sentences from 30 news articles to create an abstractive sentence compression corpus. The authors ask two annotators to compress sentences by paraphrasing while preserving the most important information and ensuring that the compressions are grammatical.

These datasets are typically used by unsupervised approaches as they are 200 times smaller in size compared to the annotated data used for training state-of-the-art supervised approaches. Filippova and Altun (2013) introduce an extractive compression dataset of 250k headline and first sentence compression pairs based on Google News[20], which they use for training a supervised compression method. Filippova et al. (2015) enlarge the Google News corpus to 2 million sentence–compression pairs using the same approach, out of which only 10,000 pairs are publicly released.[21] Similarly, Rush et al. (2015) create another large abstractive dataset of 4 million headline and first sentence compression pairs from news articles extracted from the Annotated Gigaword corpus (Napoles et al., 2012). Although these datasets are large, they predominantly address headline generation for news.

Creating such large corpora manually for a new task or domain is hard. Toutanova et al. (2016) pioneered the manual creation of a multi-reference compression dataset MSR-OANC

[18] https://www.jamesclarke.net/media/data/broadcastnews-compressions.tar.gz
[19] https://www.jamesclarke.net/media/data/written-compressions.tar.gz
[20] https://github.com/google-research-datasets/sentence-compression
[21] http://storage.googleapis.com/sentencecomp/compression-data.json

with 6k sentence–short paragraph pairs from business letters, newswire, journals, and technical documents sampled from the Open American National Corpus[22]. They provide five crowd-sourced rewrites for a fixed compression ratio and also acquire quality judgments. This dataset covers multiple genres compared to the large automatically collected compression datasets.

3.1.3 Information Recommendation

Dataset	Domain	Users	Items	Ratings	D (%)	R	S
MovieLens 1M	movie	6,040	3,706	1,000,209	4.47	no	n
MovieLens 10M	movie	71,567	10,681	10,000,054	1.31	no	n
MovieLens 20M	movie	138,493	27,278	20,000,263	0.52	no	n
Netflix	movie	480,189	17,770	100,480,507	1.17	no	n
FilmTrust (Guo et al., 2013)	movie	1,508	2,071	35,497	1.14	n	n
Yahoo	music	1,823,179	136,736	717,872,016	0.28	n	n
Last.fm	music	359,347	186,642	17,559,530	0.03	n	n
Jester (Goldberg et al., 2001)	jokes	124,113	150	5,865,235	31.50	n	n
CiteULike	academic	175,992	22,715	538,761	0.01	n	n
BibSonomy (Benz et al., 2010)	academic	4,990	432,164	1,619,210	0.08	n	n
Book-Crossing (Ziegler et al., 2005)	books	278,858	271,379	1,149,780	0.001	n	n
YOW (Zhang, 2005)	news	28	5,921	10,010	6.0	n	n
Plista (Kille et al., 2013)	news	70,353	14,897,978	84,20795	0.008	n	n
Adressa (Gulla et al., 2017)	news	15,514	923	2,717,915	0.19	n	n
Amazon (McAuley et al., 2015)	cd	75,258	64,443	1,097,592	0.02	y	y
Amazon (McAuley et al., 2015)	toy	19,412	11,924	167,597	0.07	y	y
Amazon (McAuley et al., 2015)	music	5,541	3,568	64,706	0.32	y	y
Amazon (McAuley et al., 2015)	kindle	68,223	61,934	982,619	0.02	y	y
Amazon (McAuley et al., 2015)	electronic	192,403	63,001	1,685,748	0.01	y	y
Amazon (McAuley et al., 2015)	movie	123,960	50,052	1,697,533	0.03	y	y
Yelp	restaurant	199,445	115,798	3,072,057	0.01	y	y

Table 3.3: Overview of the existing datasets for recommendation. Abbreviations and Symbols: D: Density (average number of users rated, % w.r.t all the data), R: Reviews, S: Summaries, n: not available, y: available

The most used publicly available item recommendation datasets are in the domains such as, movies, music, photos, books. The most popular dataset is the Netflix Prize dataset, a dataset released during an open competition conducted by Netflix in 2006.[23] The goal of the competition was to predict user ratings for movies, based on user's previous ratings. The competition attracted 20,000 participants across the globe and provided a platform for new state-of-the-art

[22] https://www.anc.org/data/oanc
[23] https://www.netflixprize.com/

systems. Other movie recommender datasets include MovieLens[24] by GroupLens Research and FilmTrust[25] introduced by Guo et al. (2013).

Jester is a unique joke recommendation dataset introduced by Goldberg et al. (2001), where the ratings density is the highest by an order of magnitude i.e., on an average number users rated 30% of all the jokes. MovieLens 10M and Netflix, as a comparison, have roughly 1% of the movies rated by the users.

Large recommendation datasets also exist for music domain, for example, Yahoo music[26], Last.fm[27]. Other datasets include acamedic-paper recommendation like CiteUlike[28] and Bibsonomy[29] by Benz et al. (2010), Book-Crossing by Ziegler et al. (2005). McAuley et al. (2015) introduced a large Amazon dataset for various domains and made it publicly available.[30] This dataset includes data in domains like electronics, books, music, movies, toys, kindle and many more. This is a unique dataset which also has review and summary pairs additional to the ratings. Another dataset which also has similar properties is the Yelp Dataset[31]. These datasets give us a unique opportunity to combine our information summarization and recommendation approaches to have a mutual benefit.

However, in the field of journalism i.e., news recommendation, to our knowledge there are three publicly available news datasets, YOW (Zhang, 2005), Plista (Kille et al., 2013) and, Andressa (Gulla et al., 2017). YOW is the smallest dataset but the contextual information is too less to model the user profiles. Plista is a large collection of logs from 13 German news websites, however, they do not provide explicit ratings. Andressa[32] provides a Norwegian news recommendation dataset with 923 news articles, 15,514 users and 2,717,915 ratings. However, most of these datasets do not have reviews or comments which we require to combine our information summarization approaches with the recommendation. Most of the other datasets where there is such information are proprietary of the news organizations, which obliges us to use non-news datasets to test our algorithms and can be directly transferred to news recommendation.

3.2 Live Blog Summarization Corpora

Live blogs are dynamic news articles providing a rolling textual coverage of an ongoing event. One or multiple journalists continually post micro-updates about the event, which are dis-

[24] https://grouplens.org/datasets/movielens/
[25] https://www.librec.net/datasets.html
[26] http://webscope.sandbox.yahoo.com/catalog.php?datatype=r
[27] https://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-360K.html
[28] http://konect.cc/networks/citeulike-ut/
[29] https://www.kde.cs.uni-kassel.de/wp-content/uploads/bibsonomy/
[30] http://jmcauley.ucsd.edu/data/amazon/index.html
[31] https://www.yelp.com/dataset/challenge
[32] http://reclab.idi.ntnu.no/dataset/

Title	URL
BBC	http://www.bbc.com/news/live/uk-politics-33406777
The Guardian	https://www.theguardian.com/politics/blog
KBPS	https://www.kpbs.org/news/2020/mar/26/coronavirus-live-updates/
The Telegraph	https://www.telegraph.co.uk/finance/budget/9171268/Pasty-tax-live.html
The New York Times	https://www.nytimes.com/live/paris-attacks-live-updates/
NBC News	https://www.nbcnews.com/search?q=live-blog
Independent	https://www.independent.co.uk/topic/live-blog-0
The Daily Telegraph	https://www.dailytelegraph.com.au/search-results?q=live-blog
The Sun	https://www.thesun.co.uk/topic/live-blog/
Politico	https://www.politico.eu/article/brexit-deal-live-blog/
Evening Standard	https://www.standard.co.uk/topic/live-blog
The Spectator	https://www.spectator.co.uk/article/eastleigh-by-election-live-blog
The Spinoff	https://thespinoff.co.nz/tv/05-03-2018/the-epic-spinblog-oscars-live-blog/
Business Insider	https://www.businessinsider.com/s?q=live-blog
Mirror	https://www.mirror.co.uk/news/politics/brexit-vote-march-live-mps-20652211
Aljazeera	https://www.aljazeera.com/uk-brexit-crisis-latest-updates-191022133107869.html
Manchester Evening	https://www.manchestereveningnews.co.uk/search/?q=live%20blog/
Science Business	https://sciencebusiness.net/news/brexit-live-blog-science-and-technology
LA Times	https://www.latimes.com/world/la-fg-brexit-updates-20160624-htmlstory.html
News Week	https://www.newsweek.com/brexit-live-coverage-britain-votes-leave-eu-473980
The Local Es	https://www.thelocal.es/20151220/the-spanish-general-election-live-blog
Der Spiegel	https://sportdaten.spiegel.de/fussball/bundesliga/ma8936043/liveticker/
Zeit Online	https://www.zeit.de/sport/2016-06/nordirland-deutschland-fussball-em-live
Aachener Nachrichten	https://www.aachener-nachrichten.de/suche/Liveblog/
Radio Bullets	https://www.radiobullets.com/notiziari/elezioni-usa-clinton-trump-live-maratona/
NZZ Mediengruppe	https://www.nzz.ch/sport/wm-2014/live-ticker-achterkette-1.18326614/

Table 3.4: Example live blogs from different news organizations

played in chronological order. The updates contain a wide variety of modalities and genres, including text, video, audio, images, social media excerpts, and external links. During the last five years, live-blogging emerged as a very popular way to disseminate news offered by many major news organizations, such as the *BBC*, *The Guardian*, *The New York Times*, *The Telegraph*, *NBC News*, or *Der Spiegel*. Although live-blogging is a recent trend, more and more news agencies are adopting them for publishing content, see Table 3.4.

News organizations have seen increase in usage of live blogging since the mid-2000s. Initially, they were used to cover sports events like football, or cricket. Later, several different kinds of events are regularly covered by live blogs, including elections, ceremonies, protests, conflicts, and natural disasters. Thurman and Schapals (2017, p.1) report a journalist's view that "live blogs have transformed the way we think about news, our sourcing, and everything".

Besides their timeliness, live blogs differ from common news articles by utilizing more original sources and providing information as smaller chunks, often written in a different tone than in traditional news writing (Thurman and Walters, 2013).

Figure 3.1 shows an example of a live blog on the constitution of a new Brexit committee provided by *The Guardian*.[33] Live blogs typically consist of metadata, such as date, title, and authors and a list of postings with the updated information. For larger events, journalists provide intermediate summaries shown at the top of the article. At the end of the broadcasting, a journalist usually aggregates the postings and, if available, intermediate summaries to present the most important information about the event as timelines, short texts, or bullet point lists to the users. Figure 3.2 shows an excerpt of a completed live blog by the *BBC* which consists of 360 postings (distributed over 19 pages) and a summary shown as four bullet point items.[34]

Live blogs as such have been previously discussed in the domain of digital journalism. Thorsen (2013) gives a general introduction about challenges and opportunities of live blogging. Thurman and Walters (2013) and Thurman and Newman (2014) study the production processes and the readers' consumption behavior, Thurman and Schapals (2017) evaluate aspects of transparency and objectivity, and Thorsen and Jackson (2018) analyze sourcing practices in live blogs. Further works discuss certain types of live blogs, such as live blogs on sport events (McEnnis, 2016) or terrorist attacks (Wilczek and Blangetti, 2018). None of these works focuses on intermediate or final summaries in live blogs or computational approaches to assist the journalists.

In this work, we propose to leverage these human-written summaries to investigate the novel task of automatic live blog summarization. To this end, we provide a new corpus construction approach for producing a dataset of live blogs for this new summarization task. Our work has multiple direct applications in digital journalism and news research, since automatic summarization tools for live blogs help journalists to save time during live-blogging and enable instant updates of the intermediate summaries on a live event. However, the automatic live blog summarization task also comes with new challenges:

1. Unlike a news article, the postings of a live blog do not form one coherent piece of text. Instead, each posting introduces facts or opinions from a single source which might be highly or only marginally related to the overarching topic. For example, the live blog in Figure 3.2 contains a posting commenting the relationship between Theresa May and Angela Merkel, which is related to the overall Brexit topic, but not to the Supreme Court case. In similar lines, the live blog contains multiple topic shifts (e.g., focusing on the MP's opinions or the government appeal). This lets us assume that single-document summarizers cannot be used out of the box.

[33] https://www.theguardian.com/politics/blog/live/2019/jan/07/brexit-latest-commons-vote-boris-johnson-claims-no-deal-is-closest-to-what-people-voted-for-politics-live (accessed January 7, 2019)

[34] https://www.bbc.com/news/live/uk-politics-37976580/ (accessed January 7, 2019)

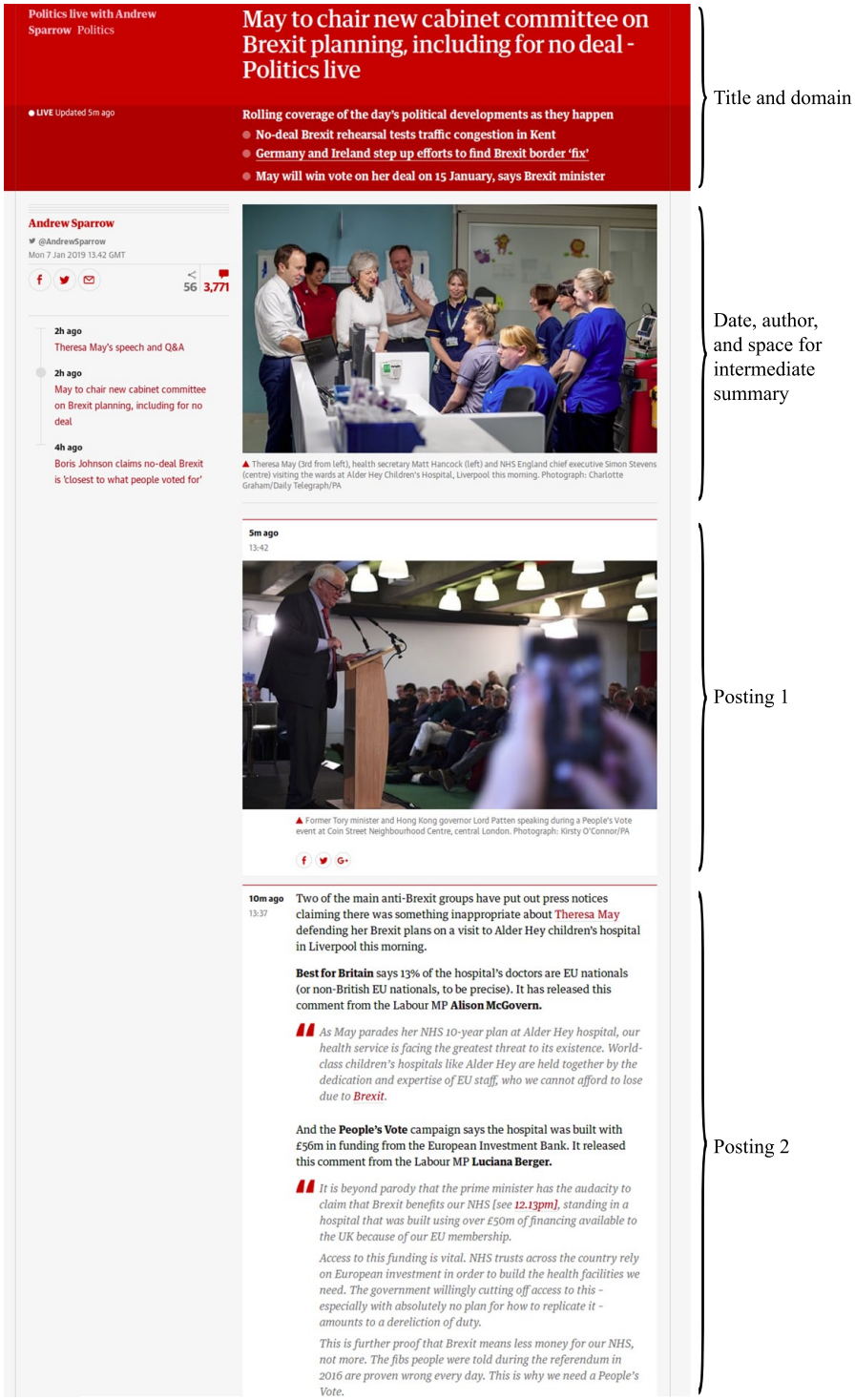


Figure 3.1: Live blog example from *The Guardian* (two newest postings visible)

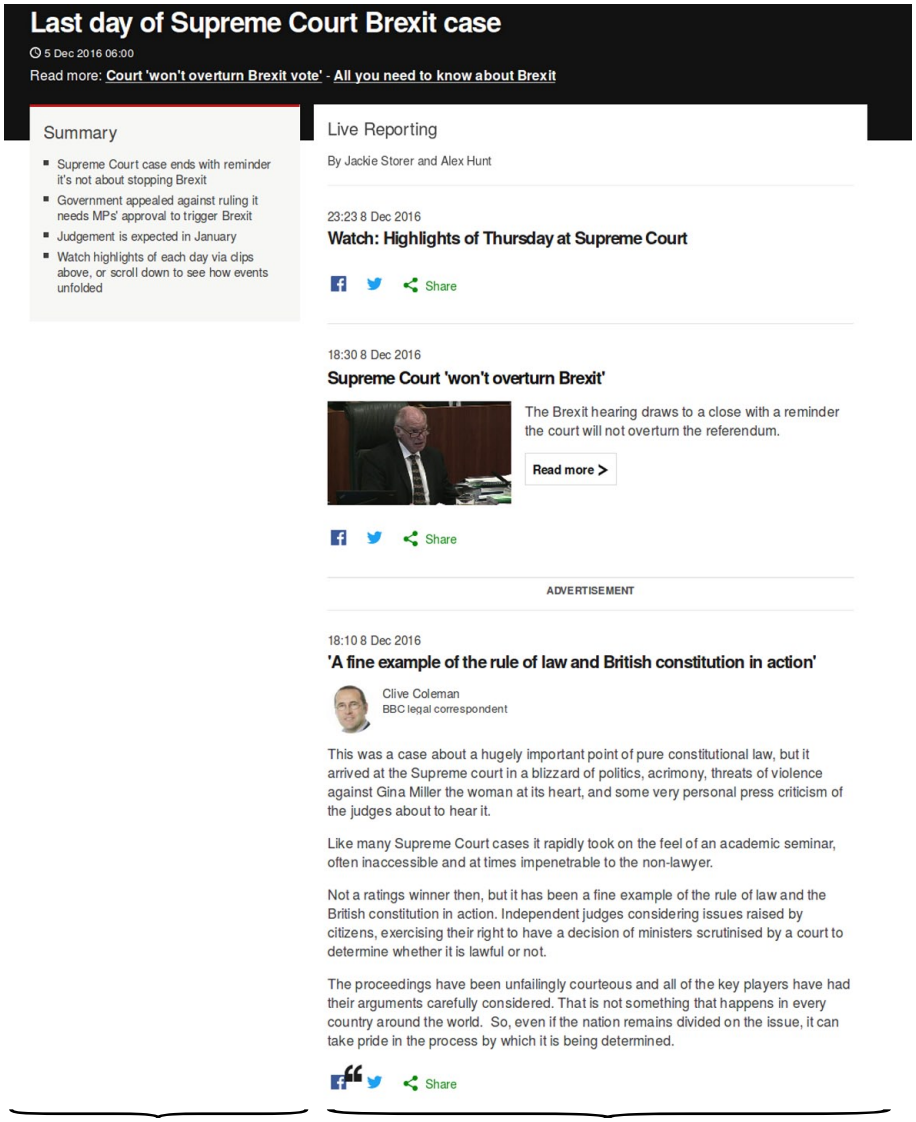


Figure 3.2: Archived live blog example from the *BBC* (three newest postings visible)

2. A particular challenge is that positional features cannot be used to estimate information importance, because live blogs are chronologically ordered and, unlike news articles, do not necessarily report the most important information first. Thus, baselines that extract the first few sentences or single-document summarization approaches building extensively on the position of a sentence are not suitable for live blog summarization.

3. The postings of live blogs are very heterogeneous, covering multiple genres, modalities, and styles. They also differ in their length and, unlike most multi-document summarization datasets, they are hardly redundant. Moreover, existing datasets contain a maximum of 20 source documents, whereas live blogs have a larger number of postings (typically more than 100) that act like individual small documents. Automatic live blog summarization approaches therefore have to deal with heterogeneous data and identify novel ways of judging importance that are not solely based on the frequency signal.

In summary, live blog summarization is a special kind of multi-document summarization, but faces highly heterogeneous, temporally ordered input. It is similar to update summarization, but has to deal with low redundancy and occasional topic shifts. Moreover, it is related to real-time summarization, where summaries are to be created without having full information about the topic yet. We will investigate these challenges in Chapter 4 Section 4.3.2, where we compare state-of-the-art multi-document summarization systems performance on live blog summarization task.

In the following subsections, we describe the live blogs, introduce the live blog summarization task and analyze the domain distribution and the heterogeneity of the corpus to understand its differences to the standard multi-document summarization like DUC'04 and TAC'08. We followed three steps to construct our live blogs summarization corpus: (1) live blog crawling yielding a list of URLs, (2) content parsing and processing, where the documents and corresponding summaries with the metadata are extracted from the URLs and stored in a JSON format, and (3) live blog pruning as a final step for creating a high-quality gold standard live blog summarization corpus.

3.2.1 Live blog Crawling

A frequently updated index webpage[35] references all archived live blogs of the Guardian. We take a snapshot of this page yielding 16,246 unique live blog URLs. In contrast, the BBC website has no such live blog archive. Thus, we use an iterative approach similar to BootCaT (Baroni and Bernardini, 2004) to bootstrap our corpus.

Algorithm 1 shows pseudo code for our iterative crawling approach, which is based on a small set of live blog URLs L_0 shown in Table 3.5. From these live blogs, we extract a set of seed terms K_0 using the 500 terms with the highest TF·IDF scores. Table 3.6 shows K_0 for

[35] http://www.theguardian.com/tone/minutebyminute

Title	URL
Politics round-up: 6 July	http://www.bbc.com/news/live/uk-politics-33406777
Over £36bn wiped off FTSE	https://www.bbc.com/news/live/business-34358976
Stormont	https://www.bbc.com/news/live/uk-northern-ireland-politics-35640347
Africa highlights	http://www.bbc.com/news/live/world-africa-35518162
Election Live - 7 April	https://www.bbc.com/news/live/election-2015-32170452
School Report Practice	http://www.bbc.com/news/live/education-31313670
IPCC report launch	https://www.bbc.com/news/live/science-environment-29820051
Junior doctor's strike	http://www.bbc.com/news/live/health-35290222
Search for Flight QZ8501	http://www.bbc.com/news/live/world-asia-30630322
Oregon shooting	http://www.bbc.com/news/live/world-us-canada-34420055

Table 3.5: Initial BBC live blogs links used to extract seed terms

world	technology	UK	business	politics	health
education	science	environment	Africa	Asia	Europe
Latin America	Middle East	US and Canada	Northern Ireland	Scotland	NHS
Nottingham	headlines	issues	justice	royal	crime
Northampton	details	risk	emergency	food	bid
Birmingham	traffic	updates	oxford	schools	commons
investment	Essex	amendment	national	officer	safety
investigation	Sheffield	appeal	jobs	rangers	residents
workers	scene	community	midlands	authority	spending
evidence	law	housing	concerns	impact	charges

Table 3.6: Sample seed terms extracted from the initial ten BBC live blogs

our corpus. The iterative procedure uses the seed terms K_0 to gather new live blog URLs by issuing automated Bing queries[36] created using recurring URL patterns P for live blogs (line 7). We collect all valid links returned by the Bing search (line 8) and extract new key terms K_t from each crawled live blog (line 12). Similar to the seed terms, we define K_t as the top 500 terms sorted by TF·IDF. The new key terms are then used to generate the Bing queries in the subsequent iterations (line 7). The process is repeated until no new live blogs are discovered anymore (line 7). For our corpus, we use the pattern

`site:http://www.bbc.com/news/live/`*‹key term›*

where *‹key term›* is one of the extracted key terms K_{t-1} from the previous iteration (or the seed terms if $t = 1$).

[36] https://azure.microsoft.com/en-us/services/cognitive-services/bing-web-search-api

Algorithm 1 Iterative Live blog crawling

input Seed URLs L_0, URL patterns P
output List of live blog URLs L
procedure CrawlLiveBlogs
    $L \leftarrow L_0$
    $K_0 \leftarrow$ extractKeyTerms(L_0)
    for $t = 1...T$ **do**
        $Q_t \leftarrow$ createQueries(K_{t-1}, P)
        $L_t \leftarrow$ obtainLinks(Q_t)
        if $L \cup L_t = L$ **then**
            return L
        else
            $K_t \leftarrow$ extractKeyTerms$(L_t) - \bigcup_{i=0}^{t-1} K_i$
            $L \leftarrow L \cup L_t$
        end if
    end for
    return L
end procedure

Using the proposed algorithm, we run 4,000 search queries returning each around 1,000 results on average, from which we collected 9,931 unique URLs. Although our method collects a majority of the live blogs in the 4,000 search queries, a more sophisticated key terms selection could minimize the search queries and maximize the unique URLs. An important point to note is that we find the collected BBC live blog URLs predominantly cover more recent years. This usage could be due to the Bing Search API preferring recent articles for the first 100 results.

By choosing a different set of seed URLs L_0 or seed terms K_0 and different URL patterns P, our methodology can be applied to other news websites featuring live blogs, such as *The New York Times*, the *Washington Post* or the German *Spiegel*.

3.2.2 Content Parsing and Processing

Once the URLs are retrieved, we fetch the HTML content, remove the boiler-plate using the BeautifulSoup[37] parser and store the cleaned data in a JSON file. During this step, unreachable URLs were filtered out. We discard live blogs for which we could not retrieve the summary or correctly parse the postings.

We parse metadata, such as URL, author, date, genre, summaries, and all postings for each live blog using site-specific regular expressions on the HTML source files. The automatic extraction is generally difficult, as the markup structure may change over time. For BBC live blogs, both the postings and the bullet-point summaries follow a consistent pattern, we can

[37] https://pypi.org/project/beautifulsoup4/

Source	Crawling	Processing	Pruning
LB-BBC	9,931	7,307	762
LB-Guardian	16,246	6,405	1,683
Total corpus	26,177	13,712	2,655

Table 3.7: Number of live blogs for BBC and the Guardian after each step of our pipeline

easily extract automatically. For the Guardian, we identify several recurring patterns which cover most of the live blogs. The Guardian provides live blogs since 2001, but they were in an experimental phase until 2008. Due to the lack of a specific structure or a summary during this experimental phase, we had to remove about 10k of the crawled live blogs, for which we could not automatically identify the postings or the summary. However, after 2008, the live blogs showed a consistent structure, as they received a prominent place in the web site. After this step, 7,307 live blogs remain for the BBC and 6,450 for the Guardian.

3.2.3 Live Blog Pruning

To further clean the data, we remove live blogs covering multiple topics, as they can be quite noisy. For example, BBC provides some live blogs discussing all events happening in a certain region within a given time frame (e.g., *Essex: Latest updates*). We also prune live blogs about sport games and live chats, because their summaries are based on simple, easy-to-replicate templates.

We further prune live blogs based on their summaries. We first remove a sentence of a summary if it has less than three words. Then, we discard live blogs whose summaries have less than three sentences. This is to ensure the quality of the corpus, since overly short summaries would yield a different summarization goal similar to headline generation and they are typically an indicator for a non-standard live blog layout in which the summary has been separated to multiple parts of the website.

After the whole pruning step, 762 live blogs remained for BBC and 1,683 for the Guardian. Overall, 10 % of the initial set of live blogs, both for BBC and the Guardian remain after our selective pruning. This is to ensure high-quality summaries for the live blogs. Although the pruning rejects 90 % of the live blogs, the size of the live blog corpus is still 20–30 times larger than the classical corpora released during DUC, TREC, and TAC tasks.

3.2.4 Corpus Analysis

Our final corpus yields a multi-document summarization corpus, in which the individual topics correspond to the crawled live blogs and the set of documents per topic corresponds to the postings of the live blog. We compute several statistics about our corpus and report them in

Statistic	LB-BBC	LB-Guardian
Number of live blogs	762	1,683
Number of postings	92,537	94,462
Average postings per live blog	95.01	56.19
Average words per posting	61.75	107.53
Average words per summary	59.48	42.23

Table 3.8: Corpus statistics for BBC and the Guardian live blogs

Domain	Live blogs	Proportion (%)
Politics	834	31.41
Business	421	15.86
General News	369	13.90
UK local events	368	13.86
International events	337	12.69
Culture	186	7.01
Science	60	2.26
Society	27	1.02
Others	53	2.00

Table 3.9: Domain distribution of our final corpus

Table 3.8. The number of postings per live blog is around 95 for BBC and 56 for the Guardian. In comparison, standard multi-document summarization datasets like DUC'04 and TAC'08 introduced in Section 3.1.1 have only 10 documents per topic. Furthermore, we observe that the postings are quite short as there is an average of 62 words per posting for BBC and 108 for the Guardian. The summaries are also shorter than the summaries of standard datasets: The summaries of DUC'04 and TAC'08 are expected to contain 100 words. However, our final corpus is larger overall, because it contains 2,655 live blogs (i.e., topics) and 186,999 postings (i.e., documents). With that many data points, machine learning approaches become readily applicable.

Domain Distribution. The live blogs in our corpus cover a wide range of subjects from multiple domains. In Table 3.9, we report the distribution across all domains in the final corpus (BBC and Guardian combined). While we observe that politics, business, and news are the most prominent domains, there is also a number of well-represented domains, such as local and international events or culture.

Heterogeneity. The resulting corpus is expected of exhibiting various levels of heterogeneity. Indeed, it contains live blogs with mixed writing styles (short and to the point vs. longer

	LB-BBC	LB-Guardian	DUC'04	TAC'08
TH_{JS}	0.5917	0.5689	0.3019	0.3188

Table 3.10: Average textual heterogeneity of our corpora compared to standard datasets

descriptive postings, informal language, quotations, encyclopedic background information, opinionated discussions, etc.). Furthermore, live blogs are subject to topic shifts which can be observed by changes in words usage.

To measure this textual heterogeneity, we use information theoretic metrics on word probability distributions like it was done before in analyzing the heterogeneity of summarization corpora (Zopf et al., 2016). Based on the Jensen-Shannon (JS) divergence, they defined a measure of textual heterogeneity *TH* for a topic T composed of documents $d_1, \ldots, d_n$ as

$$TH_{JS}(T) = \frac{1}{n} \sum_{d_i \in T} JS(P_{d_i}, P_{T \setminus d_i}) \tag{3.1}$$

Here, P_{d_i} is the frequency distribution of words in document d_i and $P_{T \setminus d_i}$ is the frequency distribution of words in all other documents of the topic except d_i. The final quantity TH_{JS} is the average divergence of documents with all the others and provides, therefore, a measure of diversity among documents of a given topic.

We report the results in Table 3.10. To put the numbers in perspective, we also report the textual heterogeneity of the two standard multi-document summarization corpora DUC'04 and TAC'08. The heterogeneity in BBC and Guardian are similar. Thus, heterogeneity of our corpus is much higher than in DUC'04 and TAC'08, indicating that our corpus contains more lexical variation inside its topics.

Compression ratio. Additional factors which determine the difficulty of the summarization task are the length of the source documents and the summary (Nenkova and Louis, 2008). The input document sizes of the BBC and the Guardian are on an average 5,890 and 6,048 words, whereas the summary sizes are only around 59 and 42 words respectively. In contrast, typical multi-document DUC datasets have a much lower compression ratio, since their input documents have on average only 700 words, while the summaries have 100 words. Thus, we expect that the high compression ratio makes live blog summarization even more challenging.

3.3 Chapter Summary

In this chapter, we reviewed datasets in information preparation in the areas of information summarization, condensation and recommendation. We reviewed datasets in text summarization, which are broadly classified into extractive (DBS) and abstractive (DUC'01, 0'2, '04). Sim-

ilarly, for text compression, we reviewed limited amount of small, medium and large datasets such as Ziff-Davis, MSR OANC, Google News. Lastly, in the area information recommendation we reviewed datasets from various domains like movie (MovieLens), jokes (Jester), restaurants (Yelp), and multi-domain Amazon dataset.

We later describe Live blogs, which are an increasingly popular news format to cover breaking news and live events in online journalism. Online news websites around the world are using this medium to give their readers a minute by minute update on an event. Good summaries enhance the value of the live blogs for a reader, but are often not available. Automatic live blog summarization is a new task with direct applications for journalists and news readers, as journalists can easily summarize the major facts about an event and even provide instant updates as intermediate summaries while the event is ongoing.

Furthermore, we suggest a pipeline to collect live blogs with human-written bullet-point summaries from two major online newspapers, the BBC and the Guardian. Our pipeline can be extended to collect live blogs from other news agencies as well, including the *New York Times*, the *Washington Post* or *Der Spiegel*. Based on this live blog reference corpus, we analyze the domain distribution and the heterogeneity of the corpus, which shows that live blog summarization poses new challenges in the field of news summarization. We discuss more about the challenges in the next chapter.

Chapter 4

Information Summarization

In this chapter, we introduce the first journalistic scenario studied in this thesis: the information summarization. First, we discuss multi-document summarization (MDS) as the corresponding prototypical task in natural language processing research. In Section 4.3, we discuss the challenges of MDS and motivate the need for putting the human in the loop. To this end, in Section 4.4, we propose our novel interactive summarization framework. In addition to that, we quantitatively and qualitatively discuss our results in Section 4.6. To conclude the chapter, we describe our interactive system demonstration Sherlock in Section 4.7 and summarize our findings in Section 4.8.

4.1 Motivation and Challenges

As we described in the previous chapters, with the rapid growth of information and broadcasting services, the amount of information has exploded exponentially on the web. One the one hand, as a journalist there is a lot of information to cover and still keep the user up-to-date with everything. On the other hand, a user does not have time to read everything.

Information summarization plays a crucial role in dealing with this information overload problem. The goal of information summarization is aggregating the most important information from a source (or multiple sources) to produce a summarized version. Summarization is a popular technique in journalism journalism and an essential part of preparing content which typically consist of creating content for the audience. With the emergence of information summarization at large scale in journalism, a high quality text summarization is imperative to effectively summarize important information. These systems can be used in journalism to automatically summarize news articles for the user.

Text summarization is the task of extracting important information from multiple input sources and present it to the user in the form of a textual summary. In NLP research, automatic summarization is a well-known task. In this chapter, we focus on a specific use case of

information summarization and text summarization in journalism, i.e., multi-document summarization (MDS). The goal of a MDS system is to discover important information and summarize from a collection of multiple documents.

Multi-document text summarization can be broadly defined as:

- Fully automatic, where automatic tools collect important content from different source (or sources) and provide a summary. For example, Google search provides summaries containing query words[38], Microsoft Word provides an Autosummary option[39], InXight[40] provides a summarizer to identify key phrases and sentences.

- Manual, where the human intelligence and judgment is used to compile the most important information. For example, summarizing genuinely original reporting could consist of manually gathering facts, observing events firsthand, and composing a brief write-up.

Fully automatic text summarization systems have a large potential in the journalistic use case, a) as they can save resources needed to create summaries from everyday news or live blogs, and b) it also saves the readers' time and keep them acquainted with the latest news updates. It has been five decades since the development of the first automatic summarization system by Luhn (1958). To encourage research in the field of automatic summarization, many tasks have been organized during the Document Understanding Conference[41] (DUC) and the Text Analysis Conference[42] (TAC) series. Since the 1950s, several algorithms have been developed (see Section 2.2), but the performance of the algorithms was limited, and it still remains an active research topic until today.

Despite a lot of research in this area, it is still a major challenge to automatically produce summaries that are on par with human-written ones. Most of the MDS systems follow an extractive approach of selecting important sentences from the source documents, whereas humans, cut and paste relevant information from texts, combine relevant related information and rephrase (Endres-Niggemeyer, 1998). To a large extent, the challenge of MDS is due to the complexity of the task: a good summary must include the most relevant information, omit redundancy and irrelevant information, satisfy a length constraint, and be cohesive and grammatical. But an even bigger challenge is the high degree of subjectivity in content selection (Rath et al., 1961; Lin and Hovy, 2002), as it can be seen in the small overlap of what is considered important by different users (see in Section 4.3). Optimizing a system towards one single best summary that fits all users, as it is assumed by current state-of-the-art systems, is highly impractical and diminishes the usefulness of a system for real-world use cases.

In this chapter, we propose a semi-automatic approach for MDS using an interactive concept-based model to assist users in creating a personalized summary based on their feedback. Our

[38] https://search.google.com

[39] https://docs.microsoft.com/en-us/previous-versions/office/office-2010/cc179199(v=office.14)

[40] http://www.inxight.com

[41] http://duc.nist.gov/

[42] http://www.nist.gov/tac/

model employs integer linear programming (ILP) to maximize user-desired content selection while using a minimum amount of user feedback and iterations. In addition to the joint optimization framework using ILP, we explore pool-based active learning to further reduce the required feedback. Although there have been previous attempts to assist users in single-document summarization described in Section 2.2.3 (Craven, 2000; Narita et al., 2002; Orăsan et al., 2003; Orăsan and Hasler, 2006), no existing work tackles the problem of multi-document summaries using optimization techniques for user feedback. Additionally, most existing systems produce only a single, globally optimal solution. Instead, we put the human in the loop and create a personalized summary that learns to better capture the users' needs and their different notions of importance.

Our proposed method and our new interactive summarization framework can be used in multiple human-in-the-loop application scenarios: as a journalistic writing aid that suggests important, user-adapted content from multiple source feeds (e.g., live blogs), as a tool for news readers, which provides a summary of news articles, as an interactive annotation tool, which highlights important sentences for the annotators, and as a medical data analysis tool that suggests key information assisting a patient's personalized medical diagnosis.

4.2 Related Work

In this section, we focus on extractive multi-document summarization (EMDS) as introduced in Section 2.2 and compare them to our work. Then, we introduce the state-of-the-art unsupervised and supervised extractive summarization systems which we use as baselines and to motivate the need to put the human in the loop.

Extractive summarization systems that compose a summary from a number of important sentences from the source documents are by far the most popular solution for MDS. This task can be modeled as a budgeted maximum coverage problem: Given a set of sentences in the document collection, the task is to maximize the coverage of the subset of sentences under a length constraint. The scoring function estimates the importance of the content units for a summary. Most previous works consider sentences as content units and try different scoring functions to optimize the summary.

One of the earliest systems by McDonald (2007) models a scoring function by simultaneously maximizing the relevance scores of the selected content units and minimizing their pairwise redundancy scores. They solve the global optimization problem using an ILP framework. Later, several state-of-the-art results employed an ILP to maximize the number of relevant concepts in the created summary: Gillick and Favre (2009) use an ILP with bigrams as concepts and hand-coded deletion rules for compression. Berg-Kirkpatrick et al. (2011) combine grammatical features relating to the parse tree and use a maximum-margin SVM trained on annotated gold-standard compressions. Woodsend and Lapata (2012) jointly optimize content selection and surface realization, Li et al. (2013) estimate the weights of the concepts using supervised

methods, and Boudin et al. (2015) propose an approximation algorithm to achieve the optimal solution. Although these approaches achieve state-of-the-art performance, they produce only one globally optimal summary which is impractical for various users due to the subjectivity of the task. Therefore, we research interactive human-in-the-loop based approaches in order to produce personalized summaries.

The following subsections introduce the approaches which we use as baseline in the remaining chapter.

Unsupervised Approaches

As discussed earlier, a typical unsupervised approach is modeled as a combinatorial problem: Given a document collection of n sentences $(s_1, .., s_i, .., s_n)$, the goal is to generate a summary S which is a subset of the document collection. The summaries are constrained on a word buget L, such that $\sum_{j=1}^{|S|} |s_j| \leq L$, where $|s_j|$ denotes the number of words in sentence s_j.

TF·IDF: Luhn (1958) scores sentences with the term frequency. Instead of using only term frequency, the inverse document frequency (TF·IDF) Sparck Jones (1972) of the words is used. The best sentences are then greedily extracted.

LexRank: Erkan and Radev (2004) construct a similarity graph $G(V, E)$ with the set of sentences V and edges $e_{ij} \in E$ between two sentences v_i and v_j if and only if the cosine similarity between them is above a predefined threshold of 0.1. Sentences are then scored according to their PageRank in G.

LSA: Steinberger and Jezek (2004) compute a dimensionality reduction of the term-document matrix via singular value decomposition (SVD). The sentences extracted should cover the most important latent topics.

KL-Greedy: Haghighi and Vanderwende (2009) minimize the Kullback-Leibler (KL) divergence between the word distributions of the summary and the documents. All the above unsupervised approaches are provided by the sumy package[43].

ICSI[44]: Gillick and Favre (2009) propose using global linear optimization to extract a summary by solving a maximum coverage problem considering the most frequent bigrams in the source documents. ICSI has been among the state-of-the-art MDS systems when evaluated with ROUGE (Hong et al., 2014). In Section 4.4.1, we introduce the ILP definition in detail, which will be used as a starting point for our interactive summarization approach.

[43] https://github.com/miso-belica/sumy

[44] https://github.com/boudinfl/sume

Supervised Approaches

Supervised learning is typically applied when we have sufficient training data to learn a summarization model. Typically, a supervised extractive summarization task is modeled as a sequence labeling problem using the formulation by Conroy and O'Leary (2001a): Given a document set containing n sentences $(s_1, .., s_i, .., s_n)$, the goal is to generate a summary by predicting a label sequence $(y_1, .., y_i, .., y_n) \in \{0, 1\}^n$ corresponding to the n sentences, where $y_i = 1$ indicates that the i-th sentence is included in the summary. The summaries are constructed with a word budget L, which enforces a constraint on the summary length $\sum_{i=1}^{n} y_i \cdot \mid s_i \mid \leq L$.

Earlier works on supervised extractive text summarization, use sentence representations using manually selected features, and trained the supervised classifiers to predict whether the sentence should be in summary. Wong et al. (2008b) propose an SVM and Naïve classifier with sentence representation using surface, content, event, and relevance features. Besides, Conroy and O'Leary (2001b) proposed a Hidden Markov Model (HMM) based summarization approach to include the sentence orders in the document. However, since the success of deep learning in NLP, neural network-based methods have gained popularity even in supervised text summarization. The neural network-based models achieve better performance compared to the traditional supervised approaches with less human supervision.

For a typical a neural network-based extractive summarization system there are two steps:

- Sentence encoders, where sentences are encoded as continuous vectors from s_i to h_i.
- Sentence extractor/ranker, which maps a sequence of sentence representations $h_{1:n} = h_1, \ldots, h_n$ and is fed to a model for selecting the sentences for the summary by making extraction decisions $y_{1:n} = y_1, \ldots, y_n$.

Sentence Encoders Recent work by Kedzie et al. (2018) used three different encoding strategies for mapping the word embeddings into a fixed-length vector.

- **Average Encoder**: The average encoder encodes sentences as an average of the word embeddings i.e., $h = \frac{1}{|s|} \sum_{i=1}^{|s|} w_i$.
- **Convolutional neural networks (CNN) Encoder**: The CNN encoder uses convolutional feature maps proposed by Kim (2014) to encode each sentence. The concatenation of all the outputs of the convolutional filter after max pooling over time is the final sentence representations h.
- **Recurrent neural network (RNN) encoder**: The RNN encoder uses a sentence embeddings from the concatenation of the final output states of a forward and backward RNN over the sentence s word embeddings (Chung et al., 2014).

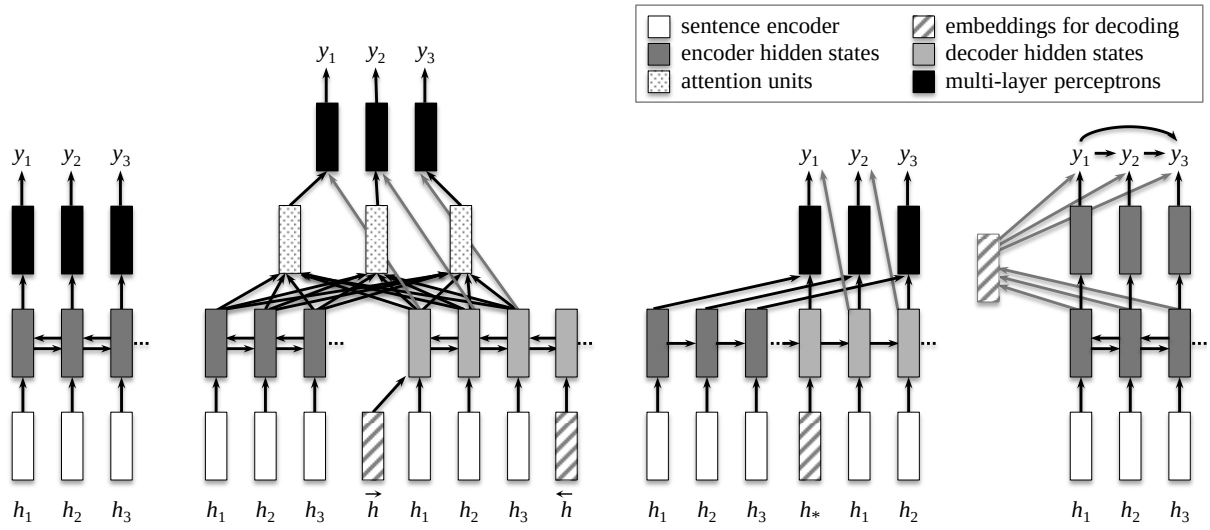


Figure 4.1: Architectures of the sentence extractors RNN, Seq2Seq, Cheng & Lapata, and SummaRuNNer

Sentence Extractors Sentence extractors take the sentence embeddings from the sentence encoders $h_{1:n} = h_1, \ldots, h_n$ and outputs extracts $y_{1:n} = y_1, \ldots, y_n$. Essentially, the sentence extractor is a classifier $p(y_{1:n}|h_{1:n})$. Figure 4.1 shows the neural network architecture of the four state-of-the-art sentence extractors we describe below.

- **RNN**: Kedzie et al. (2018) propose a simple bidirectional RNN-based tagging model. In the sentence encoder □, the forward and backward outputs ■ of each sentence are passed through a sentence selector ■ consisting of a multi-layer perceptron with sigmoid function as the output layer to predict the probability of extracting each sentence. See Figure 4.1.a for the illustration.

- **Seq2Seq**: In the same paper, Kedzie et al. (2018) also propose a sequence-to-sequence (Seq2Seq) extractor which tackles the shortcoming of the RNN extractor i.e. the inability to capture long range dependencies between the sentences. The Seq2Seq extractor thus uses an attention mechanism ⬚ popularly used in machine translation (Bahdanau et al., 2015) and abstractive summarization (See et al., 2017; Rush et al., 2015). The Seq2Seq extractor is divided into encoder ■ and decoder □, where the sentence embeddings are first encoded by a bidirectional GRU and a separate decoder GRU that transforms each sentence into a query vector. The query vector attends to the encoder output and is concatenated with the decoder GRU's output. These concatenated outputs are then fed into a multi-layer perceptron to compute the probabilities for extraction.

- **Cheng & Lapata**: Cheng and Lapata (2016) propose a Seq2Seq model where the encoder RNN ■ is fed with the sentence embedding and the final encoder state is passed on to

the first step of the decoder RNN ☐. The decoder takes the same sentence embeddings as input and the outputs are used to predict the y_i labels defining the summary. To induce dependencies of y_i on $y_{<i}$, the decoder input is weighted by the previous extraction probabilities $y_{<i}$.

– **SummaRuNNer**: Nallapati et al. (2017) propose a sentence extractor where the sentence embeddings are passed into a bidirectional RNN ■ and the output is concatenated. Then, they average the RNN output to construct a document representation, and they sum up the previous RNN outputs weighted by extraction probabilities to construct a summary representation for each time step. Finally, the extraction probabilities are calculated using the document representation, the sentence position, the RNN outputs, and the summary representation at the i-th step. The iterative summary representation process intuitively considers dependencies of y_i on all $y_{<i}$.

4.3 Limitations of Existing Solutions

In this section, we point out the limitations of the existing solutions and describe the gaps to be filled to successfully create MDS systems. We first benchmark DUC and DBS corpus, followed by discussing the challenges of the current state-of-the-art system. In the next subsection, we benchmark live blog corpus introduced in Section 3.2 and discuss the limitations of the current systems by presenting the results and analyzing system outputs.

4.3.1 Generic Summarization

In this section, we describe the experimental setup, the upper bounds we compute to benchmark DUC and DBS corpus using state-of-the-art unsupervised summarization systems. Furthermore, we discuss results and the limitations of these systems.

Experimental Setup

For our benchmark experiments, we use two different type of corpora: (1) DBS corpus (Benikova et al., 2016), and (2) DUC'04 (Over et al., 2007), for details about the corpora see Section 3.1.1. We benchmark the unsupervised systems introduced in Section 4.2 and we disregard benchmark on the supervised neural methods due to insufficient data to train them.

For evaluating the automatically created summaries against the reference summaries, we use ROUGE (Lin, 2004) with the parameters suggested by Owczarzak et al. (2012) yielding high correlation with human judgments (i.e., with stemming and without stopword removal).[45] Since DBS summaries do not have a fixed length, we use a variable length parameter L for

[45] -n 2 -m -a -x -c 95 -r 1000 -f A -p 0.5 -t 0 -2 -4

evaluation, where L denotes the length of the reference summary. All results are averaged across all topics and reference summaries and we report ROUGE-1 (R1), ROUGE-2 (R2) and ROUGE-L (RL).

Upper bound

For comparison, we compute upper bound for extractive summarization. The upper bound is retrieved by solving the maximum coverage of n-grams from the reference summary (Takamura and Okumura, 2010; Peyrard and Eckle-Kohler, 2016; P.V.S. and Meyer, 2017). Upper bound summary extraction is cast as an ILP problem, which is the core of the ICSI system. However, the only difference is that the concept weights are set to 1 if the concepts occur in the human-written reference summary. The concept extraction depends on N, which represents the n-gram concept type. In our work, we set $N = 2$ and compute the upper bound for ROUGE-2 (UB-2).

Findings

Table 4.3 shows the ROUGE scores (Lin, 2004) of multiple fully automatic extractive multi-document summarization systems in comparison to the extractive upper bound (UB-2) on DUC'04 and DBS. The results show higher scores for DBS dataset, this is due to the extractive property of the summaries as described in Section 3.1.1.

Although ICSI achieve state-of-the-art performance, its scores are still far from the extractive upper bound of individual reference summaries. The low scores are due to low inter-annotator agreement for concept selection: Zechner (2002) reports, for example, only κ = .13 and Benikova et al. (2016) κ = .23, which according to Cohen (1960) is interpreted as none to slight agreement. Most systems try to optimize for *all* reference summaries instead of personalizing, which we consider essential to capture user-desired content. Figure 4.2 illustrates the lexical overlap between a reference summary with the summary produced by the ICSI system and the extractive upper bound (UB-2). In the figure, similar concepts are marked with similar colors.

The goal of concept selection is finding the important information within a given set of source documents. Although existing summarization algorithms come up with a generic notion of importance, it is still far from the user-specific importance as shown in Figure 4.2. In contrast to fully automatic systems, humans can easily assess importance given a topic or a query. One way to achieve personalized summarization is thus by combining the advantages of both human feedback and the generic notion of importance built in a system. This allows users to interactively steer the summarization process and integrate their user-specific notion of importance. In Section 4.4, we propose a novel interactive summarization setup, which leverages user feedback to gradually personalize a summary to the user's needs.

Systems	DUC'04 R1	DUC'04 R2	DUC'04 RL	DBS R1	DBS R2	DBS RL
TF·IDF	.292	.055	.086	.377	.144	.144
LexRank	.345	.070	.108	.434	.161	.180
LSA	.294	.045	.081	.394	.154	.147
KL-Greedy	.336	.072	.104	.369	.133	.134
ICSI	.374	.090	.118	.452	.183	.190
UB-2	**.472**	**.210**	**.182**	**.848**	**.750**	**.532**

Table 4.1: ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) scores of multiple systems compared to the extractive upper bound (UB-2)

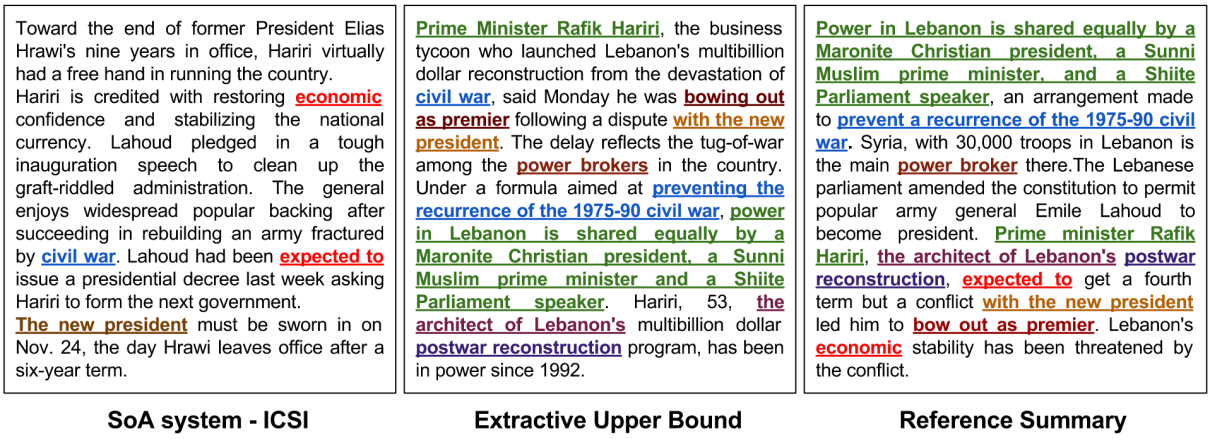


Figure 4.2: Lexical overlap of a reference summary (cluster D31043t in DUC 2004) with the summary produced by ICSI's state-of-the-art system (Boudin et al., 2015) and the extractive upper bound (UB-2)

4.3.2 Live Blog Summarization

In this section, we describe the experimental setup, the upper bounds we compute to benchmark live blogs summarization using state-of-the-art unsupervised and supervised summarization systems. Furthermore, we discuss results and the limitations of these systems.

Experimental Setup

For our benchmark experiments, we use the two live blog summarization corpora we created: (1) LB-BBC, and (2) LB-Guardian, for details about the corpora see Section 3.2. We perform experiments with the unsupervised systems used in Section 4.3.1. Furthermore, as we have sufficiently large training data, we also conduct experiment with state-of-the-art supervised summarization methods introduced in Section 4.2.

We report scores for the ROUGE metrics, and compute upper bound for ROUGE-1 (UB-1) and ROUGE-2 (UB-2) as described above in Section 4.3.1. For ROUGE, we explore two

Dataset	Train	Valid	Test
LB-BBC	610	77	75
LB-Guardian	1350	167	166

Table 4.2: Training, validation and test split sizes for LB-BBC and LB-Guardian datasets.

different summary lengths: 50 words, which corresponds to the average length of the human-written summary, and 100 words, which is twice the average length of the human-written summaries in order to give leeway for compensating the excessive compression ratio of the human-written live blog summaries.

For the supervised setup, we split the dataset into training, validation and testing consisting of 80 %, 10 %, and 10 % of the data respectively. Table 4.2 illustrates the training, validation, and test split sizes used for our experiments. We train the models to minimize the weighted negative log-likelihood over the training data D: $\mathcal{L} = -\sum_{s,y \in D} \sum_{i=1}^{n} \omega(y_i) \log p(y_i \mid y_{<i}, h)$, where $h = enc(s)$ and $enc(s)$ is the sentence encoder.

We use stochastic gradient descent with the Adam optimizer for optimizing the objective function. $\omega(y)$ represents the weights of the labels i.e. $\omega(0) = 1$ and $\omega(1) = \frac{N_0}{N_1}$[46], where N_y is the number of training samples with label y. The word embeddings were initialized using the pretrained GloVe embeddings (Pennington et al., 2014) and are not updated during training. The training is carried out for a maximum of 50 epochs and the best model is selected using an early stopping criterion for ROUGE-2 on the validation set. We use a learning rate of .0001, a dropout rate of 0.25, and bias terms of 0. The batch size is set to 32 for both LB-BBC and LB-Guardian. Additionally, due to the GPU memory limitation, the number of input sentences used by the extractors is set to 250 for LB-BBC and 200 for LB-Guardian. Lastly, we test each sentence extractor with two input encoders that compute sentence representations based on the sequence of word embeddings.

Averaging Encoder (Avg): The averaging encoder creates sentence representations

$$h_i = \frac{1}{|s_i|} \sum_{j=1}^{|s_i|} w_j$$

by averaging the word embeddings $(w_1, \ldots w_j \ldots w_{|s_i|})$ of a sentence s_i.

CNN Encoder: The CNN sentence encoder employs a series of one-dimensional convolutions over word embeddings, which is similar to the architecture proposed by Kim (2014) used for text classification. The final sentence representation h_i is the concatenation of the max-pooling

[46] a normalized weight to adjust according to the labeled classes in each dataset

	LB-BBC						LB-Guardian					
	50 words			100 words			50 words			100 words		
Systems	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
TF·IDF	.184	.030	.114	.274	.056	.155	.158	.015	.104	.245	.028	.153
LexRank	.208	.042	.132	.308	.080	.181	.198	.022	.129	.292	.039	.177
LSA	.176	.018	.018	.257	.035	.144	.143	.010	.100	.229	.020	.141
KL	.193	.032	.118	.274	.053	.160	.172	.019	.116	.256	.030	.159
ICSI	.277	.079	.180	.374	.111	.214	.223	.038	.140	.320	.050	.194
UB-1	**.439**	.184	.250	**.622**	.272	.301	**.367**	.085	.207	**.536**	.119	.269
UB-2	.419	**.230**	.263	.576	**.331**	.304	.313	**.134**	.201	.429	**.185**	.250

Table 4.3: ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (RL) scores of multiple unsupervised systems compared to the extractive upper bounds for ROUGE-1 (UB-1) and ROUGE-2 (UB-2) for summary lengths of 50 and 100 words

overtime of all the convolutional filter outputs.

Findings

Table 4.3 shows the benchmark results of the five unsupervised approaches on our live blog corpus in comparison to the standard DUC 2004 dataset. The results for 50 word summaries show that the state-of-the-art ICSI system is .15 ROUGE-1 and .2 ROUGE-2 lower than the upper bounds for LB-BBC and .1 ROUGE-1 and .1 ROUGE-2 lower for LB-Guardian's upper bounds. These differences to the upper bound are comparable to DUC 2004 as illustrated in Table 4.3. However, for LB-Guardian the upper bounds are lower in comparison to LB-BBC and DUC, which emphasizes that summaries have lower overlap with the input sources as compared to LB-BBC.

The results of our supervised approaches introduced in Section 4.2 using different extractors and encoders are shown in Table 4.4. While ICSI is the only unsupervised approach which is able to reach one-third of the upper bound, supervised approaches can reach up to 50 % of the upper bound scores. This confirms that the supervised models are able to learn importance properties of the LB-BBC dataset. However, the supervised models perform worse than ICSI on the LB-Guardian dataset. We presume this is caused by the constraint on the number of input sentences due to the GPU memory constraint.

Overall, there are improvements of about .03 ROUGE-1 and .02 ROUGE-2 when a CNN encoder is used for sentence representation as compared to the averaging encoder across all the supervised approaches, which differs from the observation by Kedzie et al. (2018). When analyzing different extractors, the Seq2Seq extractor performs best in the majority of the set-

		LB-BBC						LB-Guardian					
		50 words			100 words			50 words			100 words		
Extractor	Enc.	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
RNN	Avg.	.283	.078	.156	.379	.110	.250	.174	.019	.040	.257	.028	.062
	CNN	.296	.095	.164	.390	.123	.151	.181	.019	.040	**.273**	**.034**	**.067**
Seq2Seq	Avg.	.287	.083	.161	.380	.109	.246	.175	.020	.046	.254	.024	.060
	CNN	.296	.093	.162	**.400**	**.130**	**.261**	**.184**	**.023**	**.047**	.269	.031	.063
Cheng & Lapata	Avg.	.279	.080	.155	.372	.108	.242	.177	.020	.048	.254	.027	.061
	CNN	**.305**	**.105**	**.174**	.383	.121	.249	.181	.020	.048	.270	.030	.064
Summa RuNNer	Avg.	.245	.055	.125	.331	.067	.204	.161	.014	.030	.224	.021	.058
	CNN	.274	.080	.144	.383	.115	.248	.172	.017	.031	.256	.027	.061
UB-1	—	**.439**	.184	.250	**.622**	.272	.301	.367	.085	.207	**.536**	.119	.269
UB-2	—	.419	**.230**	.263	.576	**.331**	.304	.313	.134	.201	.429	**.185**	.250

Table 4.4: ROUGE-1 (R1), ROUGE-2 (R2), and ROUGE-L (L) scores across supervised neural methods with all extractor and encoder (enc.) pairs compared to the extractive upper bounds for ROUGE-1 (UB-1) and ROUGE-2 (UB-2)

tings, closely followed by Cheng & Lapata and RNN. SummRuNNer consistently yields lower scores across all settings.

Figure 4.3 shows the output of the best unsupervised system ICSI and the three best supervised systems (i.e. Chang & Lapata, RNN, and Seq2Seq with a CNN encoder). The outputs are compared to the extractive upper bound UB-2 and the reference summary for the BBC live blog on "Junior doctors' strike updates".[47] It can be seen that ICSI extracts sentences with the most frequent concepts (e.g., junior doctor, strike, England), but misses to identify topic shifts in the live blog's postings, such as the discussion of emergency cover. The best supervised approach Seq2Seq captures more diverse concepts (e.g, junior doctors, emergency cover, 24-hr walkout, dispute with the government) covering a greater variety of information about the strike event and its agents and reasons. However, the example also shows the challenges of live blog summarization, since most methods incorporate general statements to capture the reader's attention (e.g., "stay with us as we bring you the latest updates"), which contain little factual information, but are frequently found in the postings. Furthermore, none of the summaries provides information about the greater context and future outlook (i.e., the fact that three strikes are planned).

The above challenges however, can be easily solved by putting the human in the loop Journalists from their journalistic experience can easily identify important content of a live event and summarize them based on the contextual importance. Thus, to be able to solve such

[47] https://www.bbc.com/news//live//health-35290222 (accessed January 16, 2019)

Junior doctors in England are taking part in a 24-hour strike on Tuesday 12 January 2016 in dispute with government. Emergency cover only being provided after 08:00 GMT. There are 55,000 junior doctors – about a third of the workforce. Three strikes are planned – the last in February will see doctors refuse to provide emergency care.	This is not surprising as doctors had agreed to provide emergency care cover. There are 55,000 junior doctors in England, which is about a third of the workforce. They are taking part in a 24-hour strike in a dispute with the government over a new contract.
(a) Reference	(b) Upper bound UB-2
She also says the government's action on changing contracts was a step towards privatising the NHS. Want to know more about what's going on with the junior doctor strike in England? @twitterid thank you Noel! Here's a bit more from Jon Stanley, a junior doctor who isn't supporting the strike.	They are taking part in a 24-hour strike in a dispute with the government over a new contract. Stay with us as we bring you the latest updates, images and tweets covering the strike. Junior doctors will provide emergency cover only during the 24-hour walkout, which got under way at 08:00 GMT.
(c) ICSI	(d) Seq2Seq + CNN
This is our coverage of today's industrial action by junior doctors. Junior doctors will provide emergency cover only during the 24-hour walkout, which got under way at 08:00 GMT. Stay with us as we bring you the latest updates, images and tweets covering the strike. little do with patients - it's a middle class fight to preserve week day working - now mostly reserved for offices @twitterid @twitterid support the doctors.	They are taking part in a 24-hour strike in a dispute with the government over a new contract. Stay with us as we bring you the latest updates, images and tweets covering the strike. Tests, appointments and clinics are also being hit, and an estimated one in 10 non-emergency patients look like they will be affected on the day.
(e) Chang & Lapata + CNN	(f) RNN + CNN

Figure 4.3: System outputs on the BBC.com live blog on Junior doctors' strike updates

a complex summarization task we propose to develop human-in-the-loop-based approaches which can efficiently combine context specific importance from the journalist and the generic notion of importance from the summarization systems. The goal of such systems is to enable journalist to create summaries interactively by integrating their notion of importance in the coverage of live events. In the following section, we explain our proposed solution, an interactive summarization system which uses human in the loop to create personalized summaries.

4.4 Interactive Summarization

As discussed in the earlier sections, fully automatic EMDS approaches capture generic notion of importance and are still far from capturing user-specific importance for a given topic. To ad-

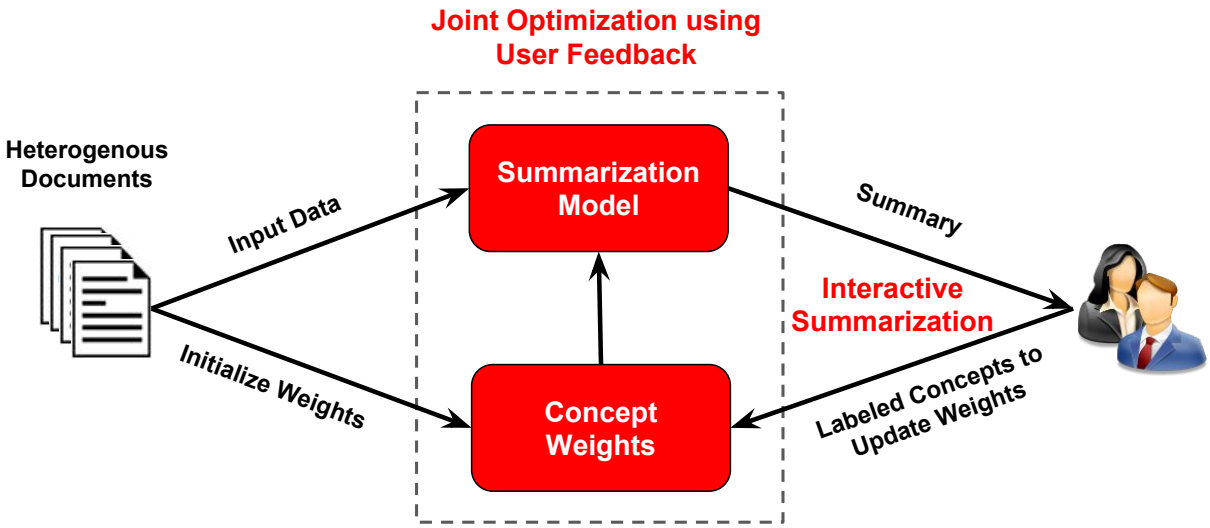


Figure 4.4: Pipeline of our interactive summarization model.

dress this research problem, we first describe our novel interactive summarization setup. Then, we introduce our joint optimization framework to learn how to update the concept weights from user feedback. Figure 4.4 illustrates the main components of our system. The user is shown a summary of the input document collection and labels all the important concepts and the unimportant concepts. Later, the weights of the labeled concepts are updated and are used by the summarization model. Thus, our proposed setup alternates the automatic creation of a summary and the acquisition of user feedback to refine the summary iteratively. And the goal of our interactive summarization setup is maximizing the user-desired content in a summary within a minimum number of iterations.

4.4.1 Summarization Model

Our starting point is the concept-based ILP summarization framework by Boudin et al. (2015) as introduced in Section 4.2. Let C be the set of concepts (e.g., bigrams, named entities) in a given set of source documents D, c_i the presence of the concept i in the resulting summary, w_i a concept's weight (e.g., tf-idf, document frequency), ℓ_j the length of sentence j, s_j the presence of sentence j in the summary, and Occ_{ij} the occurrence of concept i in sentence j.

Based on these definitions, the ILP is formulated as described in equations 4.1–4.6.

$$\begin{aligned}
&\text{Maximize} && \textstyle\sum_i w_i c_i && (4.1)\\
&\text{subject to} && \forall j. \quad \textstyle\sum_j \ell_j s_j \leq L && (4.2)\\
&&& \forall i,j. \quad s_j \, Occ_{ij} \leq c_i && (4.3)\\
&&& \forall i. \quad \textstyle\sum_j s_j \, Occ_{ij} \geq c_i && (4.4)\\
&&& \forall i. \quad c_i \in \{0,1\} && (4.5)\\
&&& \forall j. \quad s_j \in \{0,1\} && (4.6)
\end{aligned}$$

The objective function (4.1) maximizes the occurrence of concepts c_i in the summary based on their weights w_i. The two key factors for the performance of this ILP are defining the concept set C and a method to estimate the weights $w_i \in W$. Previous works have used word bigrams as concepts (Gillick and Favre, 2009; Li et al., 2013; Boudin et al., 2015) and either use document frequency (i.e. the number of source documents containing the concept) as weights (Woodsend and Lapata, 2012; Gillick and Favre, 2009) or estimate them using a supervised regression model (Li et al., 2013). For our implementation, we likewise use bigrams as concepts and document frequency as weights, as Boudin et al. (2015) report good results with this simple strategy. Our approach is, however, not limited to this setup, as our interactive approach allows for any definition of C and W, including potentially more sophisticated weight estimation methods, e.g., based on deep neural networks. In section 4.6.2, we additionally analyze how other notions of concepts can be integrated into our approach.

To involve the user into the summarization process, we propose an interactive feedback loop around this ILP formulation. Algorithm 2 provides an overview of our interactive summarization approach. The system takes the set of source documents D as input, derives the set of concepts C, and initializes their weights W. In line 5, we start the interactive feedback loop iterating over $t = 0, \ldots, T$, where T is the interaction budget (i.e. the number of rounds to query the user for feedback). We first create a summary S_t (line 6) by solving the ILP and then extract the set of concepts $Q_t \subseteq C$ from the summary (line **??**), for which we query the user in line 11. As the user feedback in the current time step, we use the concepts $I_t \subseteq Q_t$ that have been considered important by the user, i.e. the user identifies a set of important concepts from the current summary S_t, for example, by clicking on them as shown in the demonstration Section 4.7. For updating the weights W in line **??**, we may use all feedback collected until the current time step t, i.e., $I_0^t = \bigcup_{j=0}^{t} I_j$ and the set of concepts $Q_0^t = \bigcup_{j=0}^{t} Q_j$ seen by the user so far (with $Q_0^{-1} = \emptyset$). If there are no more concepts to query (i.e., $Q_t = \emptyset$), we stop the iteration and return the personalized summary S_t.

Algorithm 2 Interactive summarizer

1: **procedure** INTERACTIVESUMMARIZER()
2: **input:** Documents D
3: $C \leftarrow$ extractConcepts(D)
4: $W \leftarrow$ conceptWeights(C)
5: **for** $t = 0...T$ **do**
6: $S_t \leftarrow$ getSummary(C, W)
7: $Q_t \leftarrow$ extractConcepts$(S_t) - Q_0^{t-1}$
8: **if** $Q_t = \emptyset$ **then**
9: **return** S_t
10: **else**
11: $I_t \leftarrow$ obtainFeedback(S_t, Q_t)
12: $W \leftarrow$ updateWeights(W, I_0^t, Q_0^t)
13: **end if**
14: **end for**
15: **end procedure**

4.4.2 Joint Optimization using User Feedback

An important aspect of any interactive learning system is to enable to balance the human effort required for achieving a desired output Emamjomeh-Zadeh and Kempe (2017), i.e. querying for too much feedback could be frustrating for the users and too little would yield poor performance. Thus, we need to optimize the user feedback based summary creation process. To this end, we iteratively update the concept weights W in line 12 and jointly optimize the objective function (4.1) of the ILP setup. We define the following joint optimization models:

Accept model (ACCEPT)

This model presents the current summary S_t with highlighted concepts Q_t to a user and asks him/her to select all important concepts I_t. The weights of the concepts are updated in line 12 of Algorithm 2 by assigning the maximum weight MAX to all concepts in I_t and to weight 0 for the remaining $Q_t - I_t$ which are considered unimportant (see equation 4.7 and 4.8). The intuition behind this baseline is that the updated weights cause the ILP to prefer the user-desired concepts while avoiding unimportant ones.

$$\forall i \in I_0^t. \quad w_i = MAX \tag{4.7}$$

$$\forall i \in Q_0^t - I_0^t. \quad w_i = 0 \tag{4.8}$$

Joint ILP with User Feedback (JOINT)

Our JOINT strategy balances exploration and exploitation of the concepts for the feedback. The rationale is to acquire feedback for concepts in two phases (a) an exploration phase: where the goal is to collect feedback while exploring a large variety of concepts in the document collection, and (b) an exploitation phase: where the goal is to exploit the collected feedback. To tackle this, in our JOINT model, we change the objective function of the ILP in order to create S_t by jointly optimizing importance and user feedback. We thus replace equation (4.1) with:

$$\max \begin{cases} \sum_{i \notin Q_0^t} w_i c_i - \sum_{i \in Q_0^t} w_i c_i & \text{if } t \leq \tau \\ \sum_i w_i c_i & \text{if } t > \tau \end{cases} \tag{4.9}$$

In this JOINT model, we use the exploration phase $t = 0 \ldots \tau$ to collect feedback mostly for unseen concepts, which terminates when the user does not return any important concepts anymore (i.e., $I_t = \emptyset$). When $t \leq \tau$, equation (4.9) maximizes the use of concepts for which we yet lack feedback ($i \notin Q_0^t$) and minimizes the use of concepts for which we already have feedback ($i \in Q_0^t$). The minus term in equation (4.9) helps to reduce the score of the sentences whose concepts have received feedback already. In other words, it causes higher scores for sentences consisting of concepts which yet lack feedback. After the exploration step (i.e. $t > \tau$), we fall back to the original importance-based optimization function from equation (4.1). Lastly, weights of the concepts for which the feedback is collected are updated in the same way as the ACCEPT model.

Active learning with uncertainty sampling (AL)

Uncertainty sampling is a classic sampling technique which quantifies a classifier's uncertainty using the entropy of the predictive distribution. In our AL model, we train a classifier to predict if a concept will be accepted by the user or not and use the classifier's uncertainty to model the optimization function such that we prioritize the unseen concepts that the classifier is yet most uncertain of. The AL model employs pool-based active learning (Kremer et al., 2014) during the exploration phase in order to prioritize concepts for which the model is most uncertain. We distinguish the unlabeled concept pool $C_u = \{\Phi(\tilde{x}_1), \Phi(\tilde{x}_2), ..., \Phi(\tilde{x}_N)\}$ and the labeled concept pool $C_\ell = \{(\Phi(x_1), y_1), (\Phi(x_2), y_2), \ldots, (\Phi(x_M), y_M)\}$, where each concept x_i is represented as a d-dimensional feature vector $\Phi(x_i) \in \mathbb{R}^d$ and N, M are the number of unlabeled and labeled concepts respectively. The labels $y_i \in \{-1, 1\}$ are 1 for all important concepts in I_0^t and -1 for all unimportant concepts in $Q_0^t - I_0^t$. Initially, the labeled concept pool C_ℓ is empty, whereas the unlabeled concept pool C_u is relatively large.

The learning algorithm is presented with a $C = C_\ell \cup C_u$ and is first called to learn a decision function $f^{(0)} \colon \mathbb{R}^d \to \{-1, 1\}$, where the function $f^{(0)}(\Phi(\tilde{x}))$ should predict the label

of the input vector $\Phi(\tilde{x})$. Then, in each t th iteration during the exploration phase, where $t = 1, 2, \ldots, \tau$, the querying algorithm selects an instance $\tilde{x_t} \in C_u$ for which the learning algorithm is least certain. Thus, our learning goal of active learning is to minimize the expected loss $\mathcal{L}$ (i.e., hinge loss) with limited querying opportunities to obtain a decision function $f^{(1)}, f^{(2)}, \ldots, f^{(\tau)}$ that can achieve low error rates:

$$\min \mathbb{E}_{(\Phi(x),y)\in C_\ell}\left[\mathcal{L}(f^{(t)}(\Phi(x)), y)\right] \tag{4.10}$$

As the learning algorithm, we use a support vector machine (SVM) with a linear kernel, which has shown to generalize well in settings with little training data as compared to neural network approaches (Adel et al., 2016). To obtain the probability distribution over classes, we use Platt's calibration (Platt, 1999), an effective approach for transforming classification models into a probability distribution. Equation (4.11) shows the probability estimates for $f^{(t)}$, where $f^{(t)}$ is the uncalibrated output of the SVM in the t th iteration and a, b are scalar parameters that are learned by the calibration algorithm. The uncertainty scores are calculated as described in the equation (4.12) for all the concepts which lack feedback (C_u).

$$p(y \mid f^{(t)}) = \frac{1}{1 + \exp(af^{(t)} + b)} \tag{4.11}$$

$$u_i = 1 - \max_{y \in \{-1,1\}} p(y \mid f^{(t)}) \tag{4.12}$$

For our AL model, we now change the objective function in order to create S_t by multiplying the uncertainty scores u_i to the weights w_i. We thus replace the objective function from (4.9) with

$$\max \begin{cases} \sum_{i \notin Q_0^t} u_i w_i c_i & \text{if } t \leq \tau \\ \sum_i w_i c_i & \text{if } t > \tau \end{cases} \tag{4.13}$$

Active learning with positive sampling (AL+)

One way to sample the unseen concepts is using uncertainty as in our AL model, but another way is to model the objective function such that the predictions of the SVM classifier are also considered. The goal of this model is to prioritize those sentences which contain positively predicted concepts during optimization. Thus, in AL+, we introduce the notion of certainty $(1 - u_i)$ for the positively predicted samples ($f^{(t)}(\Phi(\tilde{x}_i)) = 1$) in the objective function (4.1) for producing S_t

$$\max \begin{cases} \sum_{i \notin Q_0^t} (1 - u_i)\ell_i w_i c_i & \text{if } t \leq \tau \\ \sum_i w_i c_i & \text{if } t > \tau \end{cases} \quad \text{where } \ell_i = \begin{cases} 0 & \text{if } f^{(t)}(\Phi(\tilde{x}_i)) = -1 \\ 1 & \text{if } f^{(t)}(\Phi(\tilde{x}_i)) = 1 \end{cases} \tag{4.14}$$

4.5 Evaluation Setup

4.5.1 Data

For our experiments, we use three different type of corpora (see Section 3.1.1 for details): (1) the DBS corpus, (2) DUC'01, DUC'02 and DUC'04, and (3) LB-BBC and LB-Guardian, the live blog summarization corpus we created in Section 3.2.

For evaluating the summaries against the reference summary we use ROUGE (Lin, 2004) with the parameters suggested by Owczarzak et al. (2012) yielding high correlation with human judgments (i.e., with stemming and without stopword removal).[48] Since DBS summaries do not have a fixed length, we use a variable length parameter L for evaluation, where L denotes the length of the reference summary. When evaluating a summarization system, it is common to report the mean ROUGE scores across clusters using all the reference summaries. However, since we aim at personalizing the summary for an individual user, we evaluate our models based on the mean ROUGE scores across clusters per reference summary.

4.5.2 Data Pre-processing and Features

To pre-process the datasets, we perform tokenization and stemming with NLTK (Loper and Bird, 2002) and constituency parsing with the Stanford parser (Klein and Manning, 2003) for English and German. The parse trees will be used in section 4.6.2 below to experiment with a syntactically motivated concept notion.

As a concept's feature representation Φ for our active learning setups AL and AL+, we use pre-trained word embeddings. We use the Google News embeddings with 300 dimensions by Mikolov et al. (2013) for English and the 100-dimensional news- and Wikipedia-based embeddings by Reimers et al. (2014) for German. Additionally, we add TF·IDF, the number of stop words, the presence of named entities, and word capitalization as features. Discrete features, such as part-of-speech tags, are mapped into the word representation via lookup tables.

4.5.3 Oracle-Based Simulation and User Study

In this section, we describe feedback collected using both simulation and user study.

Simulation

Simulation is frequently applied to evaluate interactive systems (González-Rubio et al., 2012; Knowles and Koehn, 2016; Peris and Casacuberta, 2018) for two reasons: (1) it is cost effective as compared to a user study, and (2) it is useful to have a reproducible setting to develop the systems in a theoretical and controlled environment. Therefore, we resort to an oracle-based

[48] -n 4 -m -a -x -c 95 -r 1000 -f A -p 0.5 -t 0 -2 -4 -u

approach, where the oracle is a system simulating the user by generating the feedback based on reference outputs.

To simulate user feedback in our setting, we consider all concepts $I_t \subseteq Q_t$ from the system-suggested summary S_t as important if they are present in the reference summary. Let Ref be the set of concepts in the reference summary. In the t^{th} iteration, we return $I_t = Q_t \cap Ref$ as the simulated user feedback. Thus, the goal of our system is to reach the upper bound for a user's reference summary within a minimal number of iterations. We limit our experiments to ten iterations, since it appears unrealistic that users are willing to participate in more feedback cycles. Petrie and Bevan (2009) even report only three to five iterations, however, we set the number of iterations to ten, this is to give our system to explore the large document collection for the first few iterations.

User Study

We describe a user study that we conduct with human participants to evaluate if our proposed simulated framework correlates to the real user feedback. The user study has advantage over simulation, for example, the simulation assumes perfect oracles which is unrealistic, but real user experiment is hard to reproduce and prone to noise.

The participants are instructed to generate a personalized summary using our application of the interactive summarization system demonstrated later in Section 4.7. The participants are advised to interact with our system by accepting concepts that they think are important and rejecting the concepts that they think are unimportant. The users start with the same state-of-the-art summary created using by the basic ILP (line 6 of Algorithm 3 at iteraion 0) and the users iteratively give feedback to create their personalized summary using our best system (AL). At the end, the participants were asked to create an extractive summary (i.e. S_t with t termination point in line 8) and a cohesive abstractive version of it. Figure 4.8 illustrates the instructions for the user study.

4.6 Quantitative and Qualitative Analysis

To better understand the performance of our interactive summarization framework, we organize our quantitative and qualitative analysis across five factors: (a) models (b) concept notion, (c) personalization, (d) user study and (e) scalability.

4.6.1 Analysis across models

Figure 4.6 compares the ROUGE-2 scores and the amount of feedback used over time when applied to the DBS and the DUC'04 corpus. We can see from the figure that all models show an improvement of +.45 ROUGE-2 over the baseline ICSI system after merely 4 iterations on

Usage instructions

Do a dry run on by going to the web page
http://cascade.ukp.informatik.tu-darmstadt.de/assignment.html?topic=datasets%2Fprocessed%2FDUC2004TEST%2F4doc1sum
and click the large button.

What you see next, is the analysis page. You can remind yourself of your task (1), interact with the text to tailor the summary towards your interests (2) and review the given feedback (3) before submitting it to the server.

1) A reminder about your objective
2) Interacting with the text works like a text marker: you select text using the mouse and then click Yes, No, or Erase. You can change any annotation until "submit" was clicked.
3) In the review panel, you can validate your choices, and revise them, if necessary. Once you clicked submit, the server will incorporate your feedback.

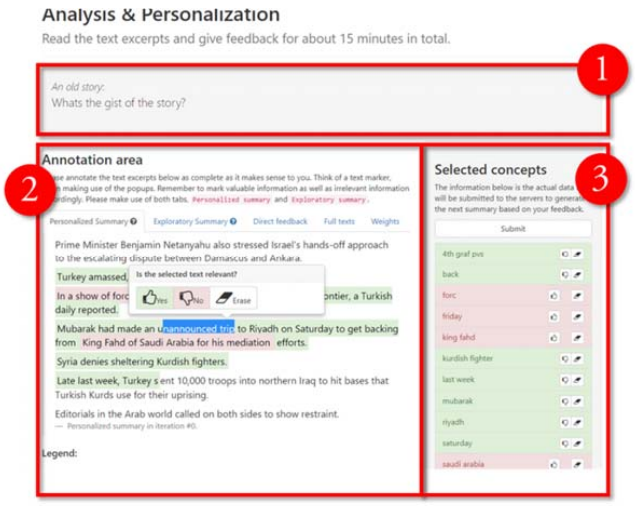


Figure 1: The analysis page

Then the interaction loop repeats: a new summary is generated (may take some time), which reflects your feedback, you interact, review, and submit.

Just try out all options, and once you feel comfortable, continue with the actual task.

PAY ATTENTENTION TO THE EXPLORATORY SUMMARY!

Although the tabs "Personalized Summary" and "Exploratory summary" have the same content in the beginning, but will diverge once you gave (enough) feedback. The personalized summary is based on your feedback, while the exploratory summary tries to be NOT like your current feedback, i.e. introduce new concepts.

Figure 4.5: Instructions of the user study with task introduction and procedure

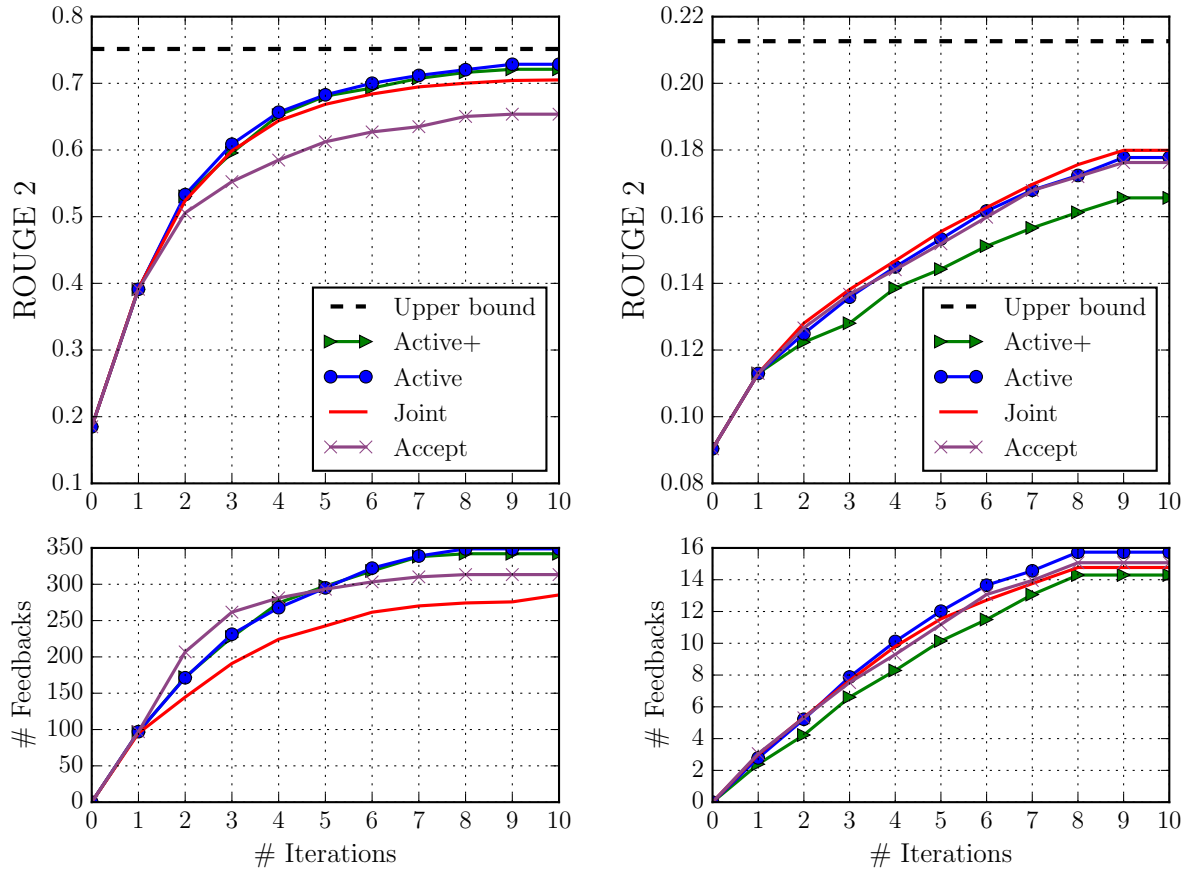


Figure 4.6: Analysis for the models over the DBS (left) and DUC'04 (right) datasets

DBS. For DUC'04, the improvements are +.1 ROUGE-2 after ten iterations, which is notable considering the lower upper bound of .21 ROUGE-2. This is primarily because DBS is a corpus of cohesive extracts, whereas DUC'04 consists of abstractive summaries. As a result, the oracles based on abstractive reference summaries provide less positive feedback, as the generated summaries have a lower overlap of concepts as compared to that of the oracles created using extractive summaries.

For DBS, it becomes clear that the JOINT model converges faster with an optimum amount of feedback as compared to other models. ACCEPT requires more feedback than JOINT, but performs low in terms of ROUGE scores. The best performing models are again AL and AL+, which reach closest to the upper bound. This is due to the exploratory nature of the models which use semantic representations of the concepts to predict uncertainty and importance of possible concepts for user feedback.

For DUC'04, the JOINT model reaches closest to the upper bound, closely followed by AL. The JOINT model consistently stays above all other models and it gathers more important concepts due to optimizing feedback for concepts which lack feedback. Interestingly, AL+

performs rather worse in terms of both ROUGE scores and gathering important concepts. The primary reason for this is the fewer feedback collected from the simulation due to the abstractive property of reference summaries, which makes the AL+ model's prediction inconsistent.

Datasets	ICSI			ACCEPT			JOINT			AL			AL+			UB-2		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
Concept Notion: Bigrams																		
DBS	.451	.183	.190	.778	.654	.453	.815	.707	.484	**.833**	**.729**	.498	.828	.721	**.500**	.848	.750	.532
LB-BBC	.277	.079	.180	.379	.191	.223	.387	.205	.230	**.394**	**.211**	**.235**	.389	.208	.231	.419	.230	.263
LB-Guardian	.223	.038	.140	.275	.105	.166	.282	.109	.175	**.299**	**.114**	**.188**	.293	.112	.182	.313	.134	.201
DUC'04	.374	.090	.118	.442	.176	.165	**.444**	**.180**	**.166**	.440	.178	.160	.427	.166	.154	.470	.212	.185
DUC'02	.350	.085	.110	.439	.178	.161	.444	.182	.165	**.448**	**.188**	.165	**.448**	.184	**.170**	.474	.216	.187
DUC'01	.333	.073	.105	.414	.171	.156	.418	.167	.149	**.435**	**.186**	**.163**	.426	.181	.158	.450	.213	.181

Table 4.5: ROUGE-1 (R1), ROUGE-2 (R2) and ROUGE-L (RL) achieved by our models after the tenth iteration of the interactive loop in comparison to the upper bound (UB-2) and the basic ILP setup. The scores in **bold** represent the model which reached closest to the upper bound.

Datasets	ACCEPT #F	JOINT #F	AL #F	AL+ #F
Concept Notion: Bigrams				
DBS	313	296	348	342
LB-BBC	20	19	22	20
LB-Guardian	14	13	15	15
DUC'04	15	14	16	14
DUC'02	14	13	15	15
DUC'01	13	11	13	13

Table 4.6: Average amount of user feedback (#F) considered by our models at the end of the tenth iteration of the interactive summarization loop

Table 4.5 shows the evaluation results of our four models to learn from user feedback based on simulation. We compare our models with ICSI as a non-interactive baseline and the extractive upper bound (UB-2). To examine the system performance based on user feedback, we analyze our models' performance on multiple datasets. The results in Table 4.5 show that our idea of interactive multi-document summarization allows users to steer a general summary towards a personalized summary consistently across all datasets. From the results, a stark difference can be seen in the performance of the models for DBS corpus, where the models start with ICSI's summary with .183 ROUGE-2 and reach close to the upper bound (UB-2) with .750 ROUGE-2. ACCEPT, JOINT, AL, and AL+ models reach .654, .707, .729 and .721 ROUGE-2 scores. The ACCEPT and JOINT models get stuck in a local optimum summary due to the

less exploratory nature of the models. Active learning based models (AL and AL+) are the best performing systems reaching closest to the upper bound (UB-2). We can see that the AL model starts from the performance of concept-based ILP summarization and nearly reaches the upper bound for all the datasets within ten iterations. In Table 4.6, we additionally evaluate the models based on the amount of feedback (#F $= |I_0^T|$) taken by the oracles to converge to the upper bound within $T = 10$ iterations. AL+ performs similar to AL in terms of ROUGE, but requires less feedback. However, JOINT model uses the least feedback as compared to rest of the models, which shows that it jointly optimizes feedback for the concepts which lack feedback.

4.6.2 The Effect of Concept Notion

Datasets	ICSI			ACCEPT			JOINT			AL			AL+			UB-2		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
Concept Notion: Content Phrases																		
DBS	.403	.135	.154	.691	.531	.430	.742	.597	.419	**.776**	**.652**	**.448**	.767	.629	.440	.848	.750	.532
LB-BBC	.277	.079	.180	.373	.189	.218	.382	.202	.224	**.390**	**.210**	**.233**	.385	.204	.228	.419	.230	.263
LB-Guardian	.223	.038	.140	.273	.101	.163	.281	.106	.171	**.295**	**.112**	**.180**	.289	.108	.178	.313	.134	.201
DUC'04	.374	.090	.118	.441	.176	.160	.441	.179	.162	**.444**	**.180**	**.162**	.422	.164	.150	.470	.212	.185
DUC'02	.350	.085	.110	.436	.181	.162	.444	.183	.165	**.446**	**.185**	**.168**	.442	.182	.162	.474	.216	.187
DUC'01	.333	.073	.105	.410	.165	.153	.417	.170	.156	**.433**	**.182**	**.161**	.420	.179	.154	.450	.213	.181

Table 4.7: ROUGE-1 (R1), ROUGE-2 (R2) and ROUGE-L (RL) achieved by our models after the tenth iteration of the interactive loop in comparison to the upper bound (UB-2) and the basic ILP setup. The scores in **bold** represent the model which reached closest to the upper bound.

Datasets	ACCEPT	JOINT	AL	AL+
	#F	#F	#F	#F
Concept Notion: Content Phrases				
DBS	110	114	133	145
LB-BBC	11	10	14	14
LB-Guardian	9	9	12	11
DUC'04	8	9	10	10
DUC'02	7	7	8	6
DUC'01	7	7	8	6

Table 4.8: Average amount of user feedback (#F) considered by our models at the end of the tenth iteration of the interactive summarization loop when the concept notion is content phrases

Our interactive summarization approach is based on the scalable global concept-based model which uses bigrams as concepts. Thus, it is intuitive to use bigrams for collecting user

feedback as well.[49] Although our models reach the upper bound when using bigram-based feedback, they require a significantly large number of iterations and much feedback to converge, as shown in Table 4.8.

To reduce the amount of feedback, we also consider content phrases to collect feedback. That is, syntactic chunks from the constituency parse trees consisting of non-function words (i.e., nouns, verbs, adjectives, and adverbs). For DBS being extractive dataset, we use bigrams and content phrases as concepts, both for the objective function in equation (4.1) and as feedback items, whereas for the DUC and live blog summarization datasets, the concepts are always bigrams for both the feedback types (bigrams/content phrases). For DUC and the LB dataset being abstractive, in the case of feedback given on content phrases, they are projected back to the bigrams to change the concept weights in order to have more overlap of simulated feedback. Table 4.8 shows that feedback based on the content phrases reduces the number of feedback by a factor of 2 across all the datasets. Furthermore, when content phrases are used as concepts for DBS, the performance of the models is lower compared to bigrams, as seen in Table 4.5 and Table 4.7.

4.6.3 User Personalization Analysis

Figure 4.7 shows the performance of different models in comparison to two different oracles i.e. two different human reference summaries for the same document cluster. For DBS, the JOINT, AL, and AL+ models consistently converge to the upper bound in four iterations for different oracles, whereas ACCEPT takes longer for one oracle and does not reach the upper bound for the other.

For DUC'04, JOINT and AL show consistent performance across the oracles, whereas AL+ performs worse than the state-of-the-art system (iteration 0) for the oracle created using abstractive summaries as shown in Figure 4.7 (right) for User:1. However, for User:2, we observe a ROUGE-2 improvement of +.1 indicating that the predictions of the active learning system are better if there is more feedback. Nevertheless, we expect that in practical use, the human summarizers may give more feedback similar to DBS in comparison to DUC'04 simulation setting.

4.6.4 User Study

We conducted the user study with 14 participants on a the document collection of topic d31043t of the DUC 2004 dataset. The results of the user study and the simulations are collected in Table 4.9. In the Table 4.9, the feedback type Manual corresponds to the feedback provided by the participants of the user study to create summary and Simulation corresponds to the summary created using the simulated feedback from the four reference summaries.

[49] We prune bigrams consisting of only functional words.

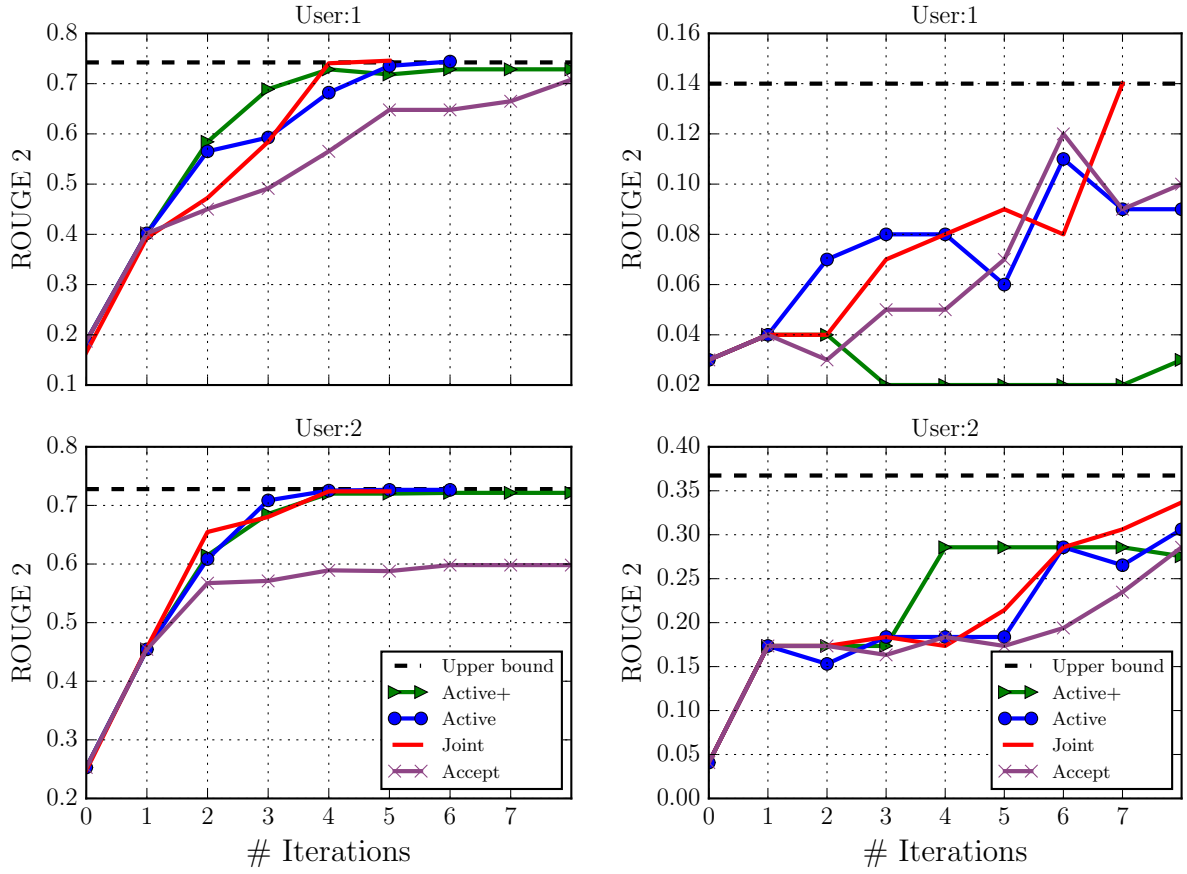


Figure 4.7: Analysis of models over cluster 7 from DBS (left) and cluster d30051t from DUC'04 (right) respectively for different oracles

Analysis of the Created Summary. Overall, the results show that the summaries created using the manual feedback took 9 iterations on average, which is similar to the simulation results. The manual feedback and the simulated feedback are also compared based on the ROUGE-2 scores of the final summary. The results show that in terms of the mean scores of ROUGE-2 the manual feedback and simulated feedback based summaries differ by a small margin +1.4% ROUGE-2, where as the maximum and minimum scores achieved by the summaries differ by a margin of +2.6% and +4.1% ROUGE-2 scores. This shows that, although some simulated feedback differ from the manual feedback, on average they are near approximate to each other.

Another key observation from our user study is that, the summaries created during the study at the end were personalized. The participants were asked to personalize their extracts by modifying the sentences which further improved the ROUGE-2 scores in some cases and decreased in some others as shown in row 2 of the Table 4.9. This is due to the sentence compression carried out by the users. The users increased the summary quality of the ex-

Feedback	Type	Number	Iterations	ROUGE-2		
				Min	Mean	Max
Manual	extract	14	9.8	.032	.063	.083
Manual	abstract	14	9.8	.000	.055	.112
Simulation	extract	4	8.6	.058	.077	.124

Table 4.9: Overview of the results using Manual and Simulated feedback in terms of ROUGE-2 scores (Min, Mean, Max), number of iterations and participants on DUC 2004 dataset topic d31043t.

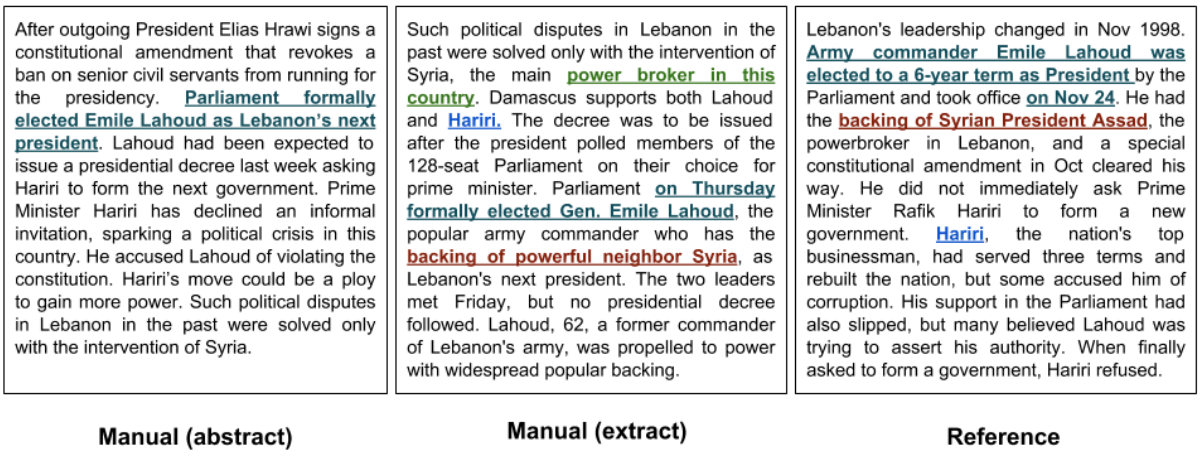


Figure 4.8: Manual (abstract) and Manual (extract) are user-created personalized summaries samples which have 0.0 and 0.083 ROUGE-2 scores in comparison to a reference summary.

tractive summary by dropping the unimportant words and rearranging the text. One of the participant's summary achieves a 0.0 ROUGE-2 score as none of the bigrams overlapped with the reference summary. Two example summaries Manual (abstract) and Manual (extract) are shown in Figure 4.8 in comparison to a reference summary. Although the Manual (abstract) summary achieves a 0.0 ROUGE-2 score, the text effectively represents the summary of the document collection. Similarly, the Manual (extract) summary which achieves the highest rouge score of 0.083 also summaries the document collection by presenting important sentences. In conclusion, the results show that our proposed simulated framework correlates to the real user feedback.

Analysis of the Types of Feedback. We also analyze the types of feedback and how it influences the quality of the summary. One key difference between the simulation and manual feedback is that the users have been explicitly told to reject the concepts (*Reject*). They have an additional option of not giving any feedback on the concepts (*None*), whereas, in the simulation, we implicitly reject all the concepts (*Reject*), which haven't been accepted (*Accept*).

Feedback		Accept			Reject			None	
	Number	Min	Mean	Max	Min	Mean	Max	Min	Mean	Max
Manual	14	15	20	46	0	20	42	20	76	136
Simulation	4	5	15	25	55	110	134	-	-	-

Table 4.10: Overview of the number of each feedback types to reach the desired summary on DUC 2004 dataset topic d31043t.

Table 4.10 shows the number of accepted (*Accept*), rejected (*Reject*), and concepts having no feedback (*None*) in both Manual and Simulation setting.

The results show that users are conservative in rejecting concepts, which is $1/5^{th}$ compared to the simulation, whereas, the simulation algorithm is over-aggressive in rejecting concepts. However, users accept more concepts compared to the simulation. Furthermore, in terms of no feedback on the concepts, we observed mixed results as some users were aggressive, and some were passive in terms of giving feedback. Overall, one key observation across both simulation and manual feedback are that accepting concepts played a considerable role in improving the summary (+0.05 ROUGE-2) compared to rejecting concepts (+0.01 ROUGE-2).

4.6.5 Scalability and Enhancements

One of the problems of the ILP-based interactive summarization model discussed in Section 4.4 is that the runtime required to provide a summary per iteration ranges from a couple of seconds for small document collections up to hours for large document collections. However, a user wants to provide feedback in an interactive manner instead of waiting for minutes or hours for the next feedback loop. In fact, a recent study by Liu and Heer (2014) has shown that a user's activity level significantly decreases even with small delays (more than 500 ms).

In joint work P.V.S. et al. (2018a), we propose an approximate model for our ILP-based summarizer that achieves interactive speeds. Therefore, we modify line 6 of Algorithm 2 to use only a sample of D_t as the input of an iteration t instead of all sentences D in the doc collect. For creating the sample D_t, two important factors play a role:

- The sample size, $K = |D_t|$ (i.e., the number of sentences in the sample), which determines the runtime of the summarization method.
- The sampling procedure, that determines *which* K sentences are part of the sample.

Sample Size

In order to determine the sample size K, we need to define an interactivity threshold (say 500 ms (Liu and Heer, 2014)) that an iteration should take maximally. Based on this threshold, we then derive a maximal K such that the resulting ILP can be solved in the given threshold.

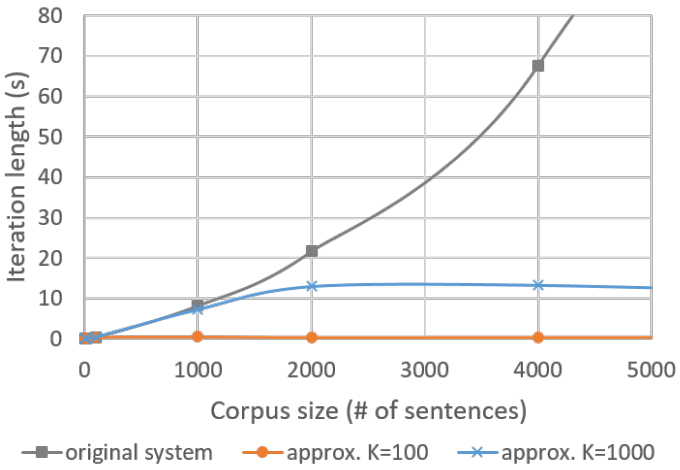


Figure 4.9: Scalability plot for our summarization system

Therefore, we determine the approximate complexity of the ILP (in number of constraints) for a given K and then estimate the runtime of solving this ILP. For estimating the runtime, we rely on calibration runs with the particular ILP solver that should be used to map a given number of constraints to an estimated runtime. We find sample size of $K = 10\% * |D|$ achieves similar results as $K = |D|$, where $|D|$ is the total number of sentences in the document collection. Figure 4.9 shows that by selecting $K = 100$, we reduce the running times to milliseconds, which normally grows exponentially when the corpus size increases.

Sampling Procedure

For deciding which sentence should be contained in the sample D_t, we propose a heuristic called information density that is computed for each sentence in D. The heuristic is the weight density of concepts normalized by the sentence length i.e., $\frac{w_i * c_i}{|s_j|}$, where w_i is the weight of the concept c_i in sentence s_j with a sentence length of $|s_j|$. For sampling, we then only select the top-K sentences based on this heuristic. The intuition is that sentences with a higher information density are more relevant to the user. It is important to note, that the information density of a sentence changes based on the user feedback since the feedback is used to update the weights of concepts and this also changes the information density of sentences that contain those concepts. For example, if all concepts of a sentence are not seen to be relevant based on the feedback, their weights will all be set to 0 and thus the information density of that sentence will also be 0.

4.7 System Applications

This section briefly looks at how interactive summarization system can be used for practical applications. We implement the interactive feedback loop, our feedback models, and the sampling procedure introduced in Section 4.6.5 in our system demonstration Sherlock [50].

We designed a web-based interface that allows users to interactively summarize document collections. Figure 4.10 illustrates a screenshot of our system, where a user is shown a summary and he/she marks all the important concepts (marked green) and the unimportant concepts (marked red). As a result, the accepted and rejected concepts appear on the right-hand side of the interface for user feedback. After the user completed his/her feedback, he/she can submit them for the next iteration. To demonstrate user interface of Sherlock, we make a video demonstration available at https://vimeo.com/257601765. In this demonstration, we show two application scenarios of Sherlock to produce personalized summaries, which we describe below.

4.7.1 Query-focused summarization

As our first scenario, imagine parents of an elementary student. The parents are worried about their child having Attention deficit hyperactivity disorder (ADHD). Their goal is to identify how ADHD is diagnosed and treated exploring a topically focused document collection. For this query-focused summarization task, we use the DUC'06 corpus (Dang, 2006). In this scenario, a parent starts exploring a preliminary summary of the document collection by giving feedback on important concepts such as 'behavioral studies', 'short attention span', 'jumpiness', 'impulsive behavior', 'stimulant medication', 'Ritalin acts on the serotonin levels in the brain', 'children who do not respond to methylphenidate', etc. They also reject unrelated concepts like 'overprescribing drugs', 'academy began developing', and 'doctors may be overdiagnosing'. The parent then reviews the accepted and the rejected concepts on the right-hand side of the user interface and submits them to let the system show an improved summary for the next iteration. Sherlock learns from this feedback and creates a new summary based on the adjusted concept weights. By using the AL model Sherlock balances between exploration and exploitation of the concepts based on the user feedback. This process continues for five to six iterations, until the parent fulfills their information need. This is achieved with the resulting summary text after all the interactions.

4.7.2 Exploratory summarization

Sherlock can also be used as a document exploration tool. In our second scenario, we deal with the exploration of large document collections (more than 1,000 documents). Sherlock's

[50] https://sherlock.ukp.informatik.tu-darmstadt.de

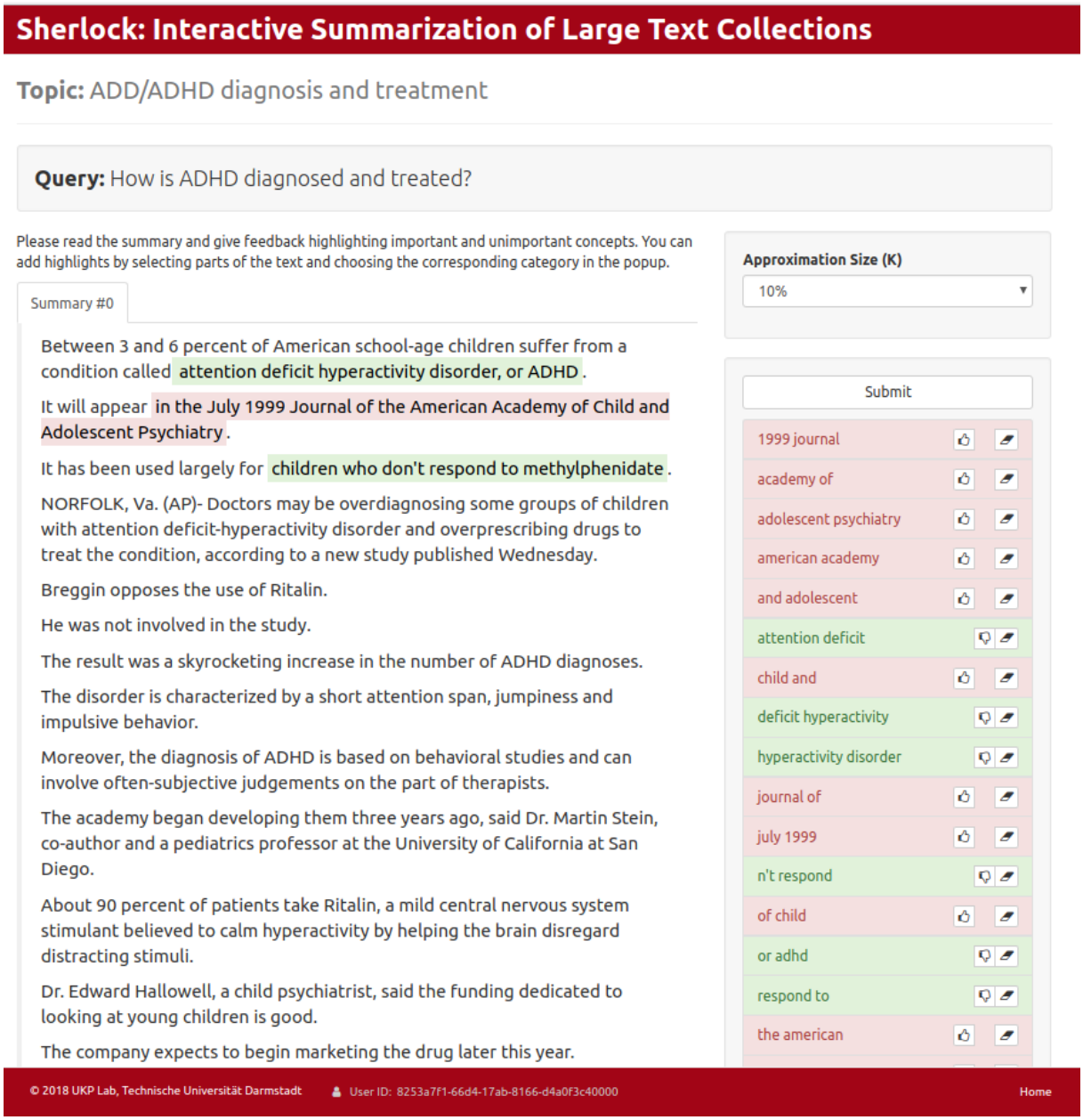


Figure 4.10: A Screenshot of Sherlock

approximation model helps to keep the runtimes low such that user's activity level is not reduced in such scenarios. Imagine a journalist investigating the situation in schools. We illustrate this scenario with the DIP corpus (Habernal et al., 2016) to explore user's information need. The journalist first explores the large collection of educational reports, web documents, and forum entries using Sherlock as his information need is yet unclear. During the first iteration, she is presented with a generic and broad summary containing some of the highlights of the issues discussed in the documents. The journalist rejects the concepts which are

not interesting for her news story, such as 'legal issues' or 'curriculum'. On the other hand, she identifies controversial topics about 'handling conflicts between children', 'bullying', or 'religious classes'. After exploring the document collection for multiple iterations, the journalist decides to write a news piece about parents' concerns about religious classes in school. Sherlock's personalized summary becomes a building block, that she extends and revises with additional facts and interviews.

4.8 Chapter Summary

In this chapter, we discuss the drawbacks of current state-of-the-art systems for information summarization specifically focusing on multi-document text summarization. Early in this chapter, we pointed out, several challenges that makes this task difficult and proposed a new framework to deal with them. Towards that end, we made the following contributions: First, we propose a novel ILP-based approach using interactive user feedback to create multi-document user-desired summaries. Using the interactive summarization setup we investigate pool-based active learning and joint optimization techniques to collect user feedback for identifying important concepts for a summary. Our models show that interactively collecting feedback consistently steers a general summary towards a user-desired personalized summary. We empirically checked the validity of our approach on standard datasets using simulated user feedback and observed that our framework shows promising results in terms of producing personalized multi-document summaries. We also conducted a user study, that showed that our proposed simulated feedback correlates to the real user feedback. In addition to that, we enhanced our interactive summarization to be able to demonstrate interactive speeds using our approximate model that bounds the runtime to keep the user's activity.

The following chapter discusses another use case of information preparation which deals with information condensation which is a natural next step of information summarization. We point out how information condensation can benefit from our human-in-the-loop framework.

Chapter 5

Information Condensation

In this chapter, we focus on data-efficient models for information condensation. We specifically focus on a text compression scenario for journalism. For text compression, the current state-of-the-art systems are based on neural networks, which require enormous amounts of data. In Section 5.1, we discuss the need for data-efficient techniques. Then, we propose our novel interactive text compression approach which uses batch mode active learning based sampling techniques to efficiently learn with minimal training data in Section 5.3.1. In Section 5.4, we evaluate our approaches experimentally using standard text compression datasets and analyze the performance of the approaches on both in-domain and out-of-domain settings. To conclude, we summarize our findings in Section 5.6.

5.1 Motivation and Challenges

Information condensation is a natural next step of information summarization. The goal of an information condensation system is to condense the aggregated important information. Condensation is essential to journalistic editing as there is more material about a topic that can be reported. In journalism, the editor's task is to condense the lengthy stories of the reporter, as the editor is the person who knows the availability of space in the newspaper. The task of the editor is to squeeze the information into the available space by deleting the unimportant information or substituting a number of words with less words. The editorial process requires deletion, organization and other modifications of the aggregated information of a news article. Some applications include creating captivating short headlines (Filippova et al., 2015) and compression of text for small screens (Corston-Oliver, 2001).

Information condensation in journalism is called *Text compression*. It is the task of condensing one or multiple sentences into a shorter text of a given length preserving the most important information. In NLP research, automatic text compression based rewriting task has attracted a lot of attention due to its simple task formulation using word deletions. An

automatic text compression can be useful in a wide range of related applications other than journalism, such as, generating captions (Wubben et al., 2016), generating automatic subtitles for television or youtube, where the rate of speech is usually higher than the rate at which text is displayed on the screen (Vandegehinste and Pan, 2004; Luotolahti and Ginter, 2015), and could be used in audio scanning devices for the blind, where a blind reader can quickly scan a document with the help of the text compression feature(Grefenstette, 1998).

Over the last few years neural *sequence-to-sequence* (Seq2Seq) models have shown remarkable success in many areas of natural language processing and specifically in natural language generation tasks, including text compression (Rush et al., 2015; Filippova et al., 2015; Yu et al., 2018; Kamigaito et al., 2018). Despite their success, Seq2Seq models have a major drawback, as they require huge parallel corpora with pairs of source and compressed text to be able to learn the parameters for the model. The training data currently used has 190,000 (Filippova and Altun, 2013) to 2 million pairs (Filippova et al., 2015). So far, the size of the training data has been proportional to the increase in the model's performance (Koehn et al., 2003; Suresh, 2010), which is a major hurdle if only limited annotation capacities are available to manually produce a corpus. That is why existing research employs large-scale automatically extracted compression pairs, such as the first sentence and the presumably shorter headline of a news article. However, such easy-to-extract source data is only available for a few, domains, genres, and language and the corresponding models do not generalize well, for example, from the headline generation of news articles to other text compression genres such as journals, books, etc. To create such large annotated corpora for a new journalistic use case would be expensive and labour intensive.

In this chapter, we propose an *interactive setup* to neural text compression, which learns to compress based on user feedback acquired during training time. For the first time, we apply *active learning* (AL) methods to neural text compression, which greatly reduces the amount of the required training data and thus yields a much more data-efficient training and annotation workflow. The goal of active learning in this chapter is different from the one used in Chapter 4. The AL-based sampling here has to actively find the best subset of examples to be annotated by the users so that the model learns efficiently with minimal data, whereas in Chapter 4, AL-based sampling is used to sample uncertain model features given a fixed query budget, whose weights are updated in the interactive setup. In our experiments, we find that this AL-based approach enables the successful transfer of a model trained on headline generation data to a general text compression task with a minimum of parallel training instances.

The objective of AL is to efficiently select unlabeled instances that a user should annotate to advance the training. A key component of AL is the choice of the *sampling strategy*, which curates the samples in order to maximize the model's performance with a minimum amount of user interaction. Many AL sampling strategies have proven effective for human-supervised natural language processing tasks other than compression (Hahn et al., 2012; Peris and Casacuberta, 2018; Liu et al., 2018b).

In our work, we exploit the application of uncertainty-based sampling using attention dispersion and structural similarity for choosing samples to be annotated for our interactive Seq2Seq text compression model. We employ the AL strategies for (a) learning a model with a minimum data, and (b) adapting a pretrained model with few user inputs to a new domain.

In the remaining sections, we first discuss related work and introduce a state-of-the-art Seq2Seq architecture for the neural text compression task as our starting point. Then, we propose our novel interactive compression approach and demonstrate how batch mode AL can be integrated with neural Seq2Seq models for text compression. In section 5.4, we introduce our experimental setup, and evaluate our AL strategies. We find that our approach successfully enables (a) learning the Seq2Seq model with only a small fraction of the training data and (b) transferring a pretrained headline generation model to a new compression task and dataset with minimal user interaction.

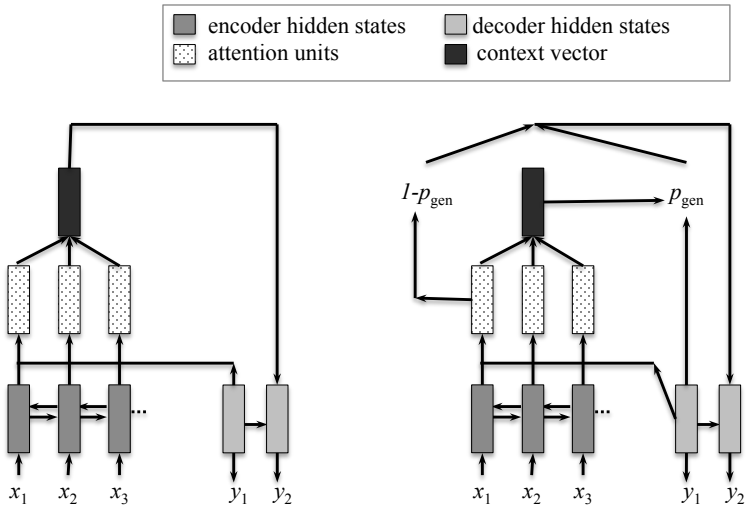
5.2 Related Work

In this section, we first focus on text compression from Section 2.3 and discuss their shortcomings. Then, we introduce the state-of-the-art Seq2Seq text compression models which we use later in this chapter in more detail.

5.2.1 Neural text compression

Filippova et al. (2015) show that Seq2Seq models without any linguistic features have the ability to delete unimportant information. Kamigaito et al. (2018) incorporate higher-order dependency features into a Seq2Seq model and report promising results. Rush et al. (2015) propose an attention-based Seq2Seq model for generating headlines. Chopra et al. (2016) further improve this task with recurrent neural networks. Although Seq2Seq models show state-of-the-art results on different compression datasets, there is yet no work which investigates whether large training corpora are needed to train neural compression models and if there are efficient ways to train and adapt them to other datasets with few annotations.

An abstractive text compression can be formalized as: given an input sentence x consisting of a sequence of N words $x_1, x_2, ..., x_N$ comprising of a fixed vocabulary V of size $\mid V \mid$, a compression are a shortened text $y = y_1, y_2, ..y_M$ of length $M < N$. Unlike related tasks like machine translation, the output length of the summary M is known to the system and is fixed before generation, which is the compression rate of the task. The goal of the system is to find an optimal sequence of summary y from Y i.e. $\text{argmax}_{y \in Y} s(x, y)$, where Y is a set of all possible sentences of length M and $s : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ is a scoring function. The scoring function is typically modeled as a local conditional distribution and this function varies for different architectures.



(a) Seq2Seq-gen (b) Pointer-gen

Figure 5.1: Architectures of the text compression neural text compression systems Seq2Seq-gen and Pointer-gen

In this work, we employ state-of-the-art Seq2Seq models with attention (Seq2Seq-gen) Rush et al. (2015) and pointer-generated networks with coverage (Pointer-gen) See et al. (2017) illustrated in Figure 5.1 as our base models, which we use for our interactive text compression setup.

Seq2Seq-gen: Rush et al. (2015) propose a Seq2Seq which is built upon the encoder-decoder framework by Sutskever et al. (2014). The encoder encodes the input sequence $\mathrm{x} = (x_1, x_2.., x_n)$ represented by an embedding matrix into a continuous space using a bidirectional LSTM network and outputs a sequence of hidden states. The decoder is a conditional bidirectional LSTM network with attention distribution (Luong et al., 2015)

$$a_i^j = \frac{\exp(e_i^j)}{\sum_{k=1}^{n} \exp(e_k^j)} \tag{5.1}$$

where e_i^j is computed at each generation step j with the encoder states h_i^{enc} and the decoder states h_j^{dec}:

$$e_i^j = q \cdot \tanh(W_h^{\mathrm{enc}} h_i^{\mathrm{enc}} + W_h^{\mathrm{dec}} h_j^{\mathrm{dec}} + b_{\mathrm{att}}) \tag{5.2}$$

where q, W_h^{enc}, W_h^{dec} and b_{att} are learnable parameters. The attention distribution a_i^j is used to compute the weighted sum of the encoder hidden states, also known as the context vector

$$c_j^* = \sum_i^n a_i^j h_i^{\text{enc}} \tag{5.3}$$

To obtain the vocabulary distribution P_j^{vocab} at generation step j, we concatenate the fixed context vector with the decoder state h_j^{dec} and pass it through two linear layers:

$$P_j^{\text{vocab}} = \text{softmax}(W_v(W_v'[h_j^{\text{dec}}; c_j^*] + b_v') + b_v) \tag{5.4}$$

where W_v, W_v', b_v and b_v' are learnable parameters. P_j^{vocab} is a probability distribution over all words in the vocabulary V. Based on the vocabulary distribution, the model generates the target sequence $\text{y} = y_1, y_2, \ldots, y_m, m \leq n$ with

$$y_j = \text{argmax}_w P_j^{\text{vocab}}(w),\ w \in V \tag{5.5}$$

for each generation step j.

Finally during training, we define the loss function for generation step j as the negative log likelihood of the target word y_j and the overall loss function for the target word sequence as $\mathcal{L}$:

$$\mathcal{L} = \frac{1}{m} \sum_{j=0}^{m} -\log P_j^{\text{vocab}}(y_j) \tag{5.6}$$

Pointer-gen: Another state-of-the-art approach we use for our experiments is the pointer-generator networks proposed by See et al. (2017). This model uses a pointer-generator network that determines a probability function to generate the words from the vocabulary V or copy the words from the source text by sampling from the attention distribution a_i^j as shown in Eq. 6.11. The model achieves this by calculating an additional generation probability p_{gen} for generation step j, which is calculated from the context vector c_j^*, the decoder state h_j^{dec}, and the current input to the decoder $x_j^{'}$:

$$p_{\text{gen}} = \sigma(W_c^T c_j^* + W_{h^{\text{dec}}}^T h_j^{\text{dec}} + W_{x'}^T x_j' + b_{\text{gen}}) \tag{5.7}$$

$$P_j(w) = p_{\text{gen}} P_j^{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i=0}^{n} a_i^j \tag{5.8}$$

where vectors W_c, $W_{h^{\text{dec}}}$, $W_{x'}$, b_{gen} are learnable parameters, n is the number of words in the source text and σ is the sigmoid function.

The model also uses an extra feature of coverage to keep track of words generated by the model to discourage repetition. In the coverage model, a coverage vector is calculated which is the sum of the attention distribution across all the previous decoding steps and it is passed on as an extra input to the attention mechanism:

$$c_i^j = \sum_{k=0}^{j-1} a_i^k \tag{5.9}$$

$$e_i^j = q \cdot \tanh(W_h^{\text{enc}} h_i^{\text{enc}} + W_h^{\text{dec}} h_j^{\text{dec}} + W_c c_i^j + b_{\text{att}}) \tag{5.10}$$

where W_c is an additional learnable parameter.

5.2.2 Active learning for data efficiency

There is a need for data-efficient models, as creating a large labeled training data is expensive and time-consuming for real-world application (e.g., text compression). Active learning (Settles, 2012) is a popular data-efficient learning technique which primarily aims to minimize the amount of user annotation efforts required for a supervised learning model.

Active learning has been researched in various real-world NLP applications in the area of data-efficient machine learning, such as text classification (Lewis and Gale, 1994), information extraction Settles and Craven (2008), cancer diagnosis (Liu, 2004), machine translation (Haffari and Sarkar, 2009), language generation (Mairesse et al., 2010), and many more. The most commonly used active learning strategy is the uncertainty sampling (Lewis and Gale, 1994). The basic intuition is that the model can skip querying for those instances which it is confident about, and focus its attention on the unlabeled instances which it is uncertain.

Uncertainty sampling has been used for many probabilistic structured prediction data-efficient models, such as, predicting the most likely sequence in sequence labeling (Culotta and McCallum, 2005) or sequence generation tasks (Mairesse et al., 2010). Only recently, these sampling methods have been applied to neural models: Wang et al. (2017a) propose an AL approach for a black box semantic role labelling (SRL) model where the AL framework is an add-on to the neural SRL models. Peris and Casacuberta (2018) use AL in neural machine translation. They propose quality estimation sampling, coverage sampling, and attention distraction sampling strategies to query data for interactive machine translation. Liu et al. (2018b) additionally propose an AL simulation trained on a high-resource language pair to transfer their model to low-resource language pairs. In another line of research, Sener and Savarese (2018) discuss a core-set AL approach as a batch sampling method for neural image classification based on convolutional neural networks. Although AL techniques have been widely used in natural language processing, to our knowledge, there is yet no work on the use of AL for neural text compression. We fill this gap by putting the human in the loop to learn

effectively from a minimal amount of interactive feedback and for the first time, we explore this data-efficient AL-based approach to adapt a model to a new compression dataset.

5.3 Interactive Text Compression

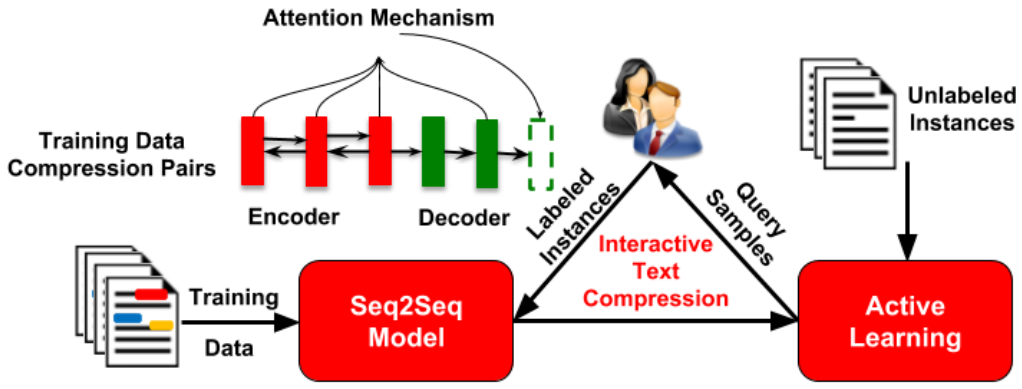


Figure 5.2: Pipeline of our interactive text compression model. The pipeline is divided into three main components: (1) Neural Seq2Seq text compression model, (2) interactive text compression, and (3) active learning

Toutanova et al. (2016) show that Seq2Seq models, which perform well on large news headline generation datasets, fail to achieve good performance on their MSR-OANC multi-genre compression dataset as compared to traditional baselines, such as ILP, due to the difference in genre and domain. The results in Toutanova et al. (2016) show that on a human rating scale of 1-3, Seq2Seq models have an average rating of 1.57 in terms of meaning and grammar as compared to ILP's 2.25. A major issue with training Seq2Seq models is the lack of domain-specific data and the expensive process to create parallel compression pairs. It is therefore indispensable to minimize the cost of data annotation. Thus, we propose to use AL whose key element is to find a strategy for selecting samples the user should annotate which yield a more efficient training process. For text compression, we suggest AL strategies to maximize the model's coverage and the diversity of the samples. Thus, to address problem of data efficiency in Seq2Seq text compression models, we first describe a novel interactive neural text compression setup. Then, we introduce our active learning strategies to select the training samples interactively for in-domain training as well as for domain adaptation. Figure 6.2 illustrates the main components of our system.

5.3.1 Interactive Text Compression

In this subsection, we introduce our interactive text compression setup. Our goal is to select the batch of samples for training efficiently with minimal samples and to become able to transfer the models to new datasets for different domains and genres with few labeled data.

We consider an initial collection of parallel instances $D = \{(\mathrm{x}_i, \mathrm{y}_i) \mid 1 \leq i \leq N\}$ consisting of pairs of input text x_i and their corresponding compression y_i. Additionally, we consider unlabeled instances $D' = \{\mathrm{x}_i \mid i > N\}$, for which we only know the uncompressed source texts. Our goal is to sample sets of unlabeled instances $S_t \subset D'$ which should be annotated by a user in each time step t. The interactive compression model can only see the labeled pairs from the initial dataset D in the beginning, but then incrementally learns from the user annotations.

Algorithm 3 provides an overview of our interactive compression setup. The inputs are the labeled compression pairs D and the unlabeled source texts D'. D is used to initially train the neural text compression model M in line 4. In our research we use Seq2Seq-gen and Pointer-gen networks introduced in Section 5.2.1, which can be replaced with any attention-based model. In line 6, we start the interactive feedback loop iterating over $t = 0, \ldots, T$. We first sample a set of unlabeled source texts S_t (line 6) by using our AL strategies introduced in section 5.3.2 and then loop over each of the unlabeled samples to be annotated or supervised by the human in line 10. As the user feedback in the current time step of sample S_t, we obtain the compressions Y_t of the sampled source texts S_t from the user and use them for online training of the model M. After T iterations or if there are no samples left for querying (i.e., $S_t = \emptyset$), we stop the iteration and return the updated Seq2Seq model M.

Algorithm 3 Interactive Text Compression

```
1:  procedure InteractiveCompression()
2:      input: Text Compression Pairs D,
3:             Unlabeled Text D'
4:      M ← learnSeq2Seq(D)
5:      for t = 0, ..., T do
6:          S_t ← getSample(D')
7:          if S_t = ∅ then
8:              return M
9:          else
10:             Y_t ← queryUser(S_t)
11:             M ← update(M, S_t, Y_t)
12:             D' ← D' − S_t
13:         end if
14:     end for
15: end procedure
```

The core component of our interactive text compression setup is the sampling function in line 6. In the following section, we discuss active learning sampling strategies and how they are applied in this interactive text compression setup.

5.3.2 Active Learning for Sampling

In this chapter, as illustrated in line 5 of Algorithm 3, the interactive text compression algorithm at each learning cycle t, iteratively queries the user for annotations of unlabeled samples S_t. The sampling is based on active learning, where those unlabeled examples are selected for which the model M is least confident on its compression. In most active learning research, samples are selected one at a time (Cohn et al., 1994; Guo and Greiner, 2007), however, in our use case, as the time required to update the model is slow and expensive, we use pool-based active learning (Brinker, 2003; Xu et al., 2007).

In active learning research, uncertainty sampling (Lewis and Gale, 1994), a popular selective sampling technique has been applied to many NLP applications such as machine translation (Peris and Casacuberta, 2018; Wang et al., 2017b), text classification (Zhu et al., 2008), and named entity recognition (Shen et al., 2004). The uncertainty sampling measure works on the principle of measuring the uncertainty of an unlabeled exampled given a model. In this section, we build upon work in uncertainty sampling based on a coverage constraint by Peris and Casacuberta (2018) and Wang et al. (2017b), and we propose a new diversity constraint uncertainty sampling to predict the sample diversity at a structural level.

Coverage constraint sampling (Coverage-AL)

As mentioned previously, the motivation behind uncertainty sampling is to find unlabeled samples for which the model is most confused while producing a compressed text. And an important uncertainty measurement on which text compression models are evaluated is the coverage Marsi et al. (2010). *Coverage* can be defined as the text compression models being able to learn the deletion or generation rules from the training samples and apply them on an input source text. Wu et al. (2016) first proposed the idea of using attention weights to calculate coverage penalty for active learning based machine translation systems. The attention weights were further extended by Peris and Casacuberta (2018) to estimate an attention dispersion based uncertainty score for a sentence. The idea of attention dispersion is that if the neural Seq2Seq compression model is uncertain then the attention weights will be dispersed across the source text while generating the target words. The samples with higher dispersion will have their attention weights uniformly distributed across the source sentences. Thus, the goal is to find the samples with high uncertainty based on attention dispersion. As we want to define the extent to which the attention distribution differs from a normal distribution, we propose to use a *skewness score*. The *skewness score* calculates the attention dispersion while

decoding a target word y_j.

$$\text{skewness}(y_j) = \frac{\frac{1}{n}\sum_{i=1}^{n}(a_i^j - \frac{1}{n})^3}{(\frac{1}{n}\sum_{i=1}^{n}(a_i^j - \frac{1}{n})^2)^{3/2}} \tag{5.11}$$

a_i^j is the attention weight assigned by the attention layer to the i-th source word when decoding the j-th target word and $\frac{1}{n}$ is the mean of the attention weights of the target word y_j.

The skewness for a normal distribution is zero, and since we are interested in the skewness of samples with heavy tails, we take the negative of the skewness averaged across all target words to obtain the uncertainty coverage score C_{score}.

$$C_{\text{score}}(\text{x}, \text{y}) = \frac{\sum_{j=1}^{m} -\text{skewness}(y_j)}{m} \tag{5.12}$$

where m is the number of target words.

Diversity constraint sampling (Diversity-AL)

Diversity sampling methods have been used in information retrieval (Xu et al., 2007) and image classification (Wang et al., 2017b). The core idea is that samples that are highly similar to each other typically yield little new information and thus low performance. Similarly, to increase the diversity of the samples in neural text compression, we propose a novel scoring metric to measure the diversity of multiple source texts at a structural level. Our intuition is that integrating part-of-speech, dependency and named entity information is useful for text compression, e.g., to learn which named entities are important and how to compress a wide range of phrase types and syntactically complex sentences. Thus, we consider part of speech tags, dependency trees, and named entity embeddings and calculate the structural similarity of the source text with regard to the target text. We use a multi-task convolutional neural network similar to Søgaard and Goldberg (2016) trained on OntoNotes and Common Crawl to learn the structural embeddings consisting of tag, dependency and named entity embeddings. The diversity score D_{score} is calculated using the cosine distance between the average of the structural embeddings of the words in the source sentence and the average of the structural embeddings of the words in the target compression as in Eq. 5.13:

$$D_{\text{score}}(\text{x}, \text{y}) = \frac{\text{E}_{\text{struc}}(\text{x}) \cdot \text{E}_{\text{struc}}(\text{y})}{||\text{E}_{\text{struc}}(\text{x})|| \cdot ||\text{E}_{\text{struc}}(\text{y})||} \tag{5.13}$$

where $\text{E}_{\text{struc}}(\cdot)$ is the average structural embedding of a text.

These AL sampling strategies are applied interactively while training to make better use of the data by selecting the most uncertain instances. Additionally, both strategies can be applied

for domain adaptation by actively querying user annotations for a domain-specific dataset in an interactive text compression setup, which we describe next.

5.4 Evaluation Setup

For our experiments, we use the large Google News text compression corpus[51] by Filippova and Altun (2013). Recent studies on text compression have extensively used this dataset (e.g., Zhao et al., 2018; Kamigaito et al., 2018). We carry out in-domain active learning experiments on the Google News compression corpus. This dataset is divided into 195,000 training, 5,000 development and 10,000 test data. To evaluate our interactive setup, we adapt the trained models to the MSR-OANC text compression corpus by Toutanova et al. (2016). This corpus is well-suited to evaluate our interactive setup, since it is sourced from mixture of newswire, letters, journals, and non-fiction genres, in contrast to the Google News corpus covering only newswire. This dataset is divided into 5,000 training, 448 development and 785 test data, for transfer learning. To preprocess the datasets, we perform tokenization. We obtain the structural embeddings for a sentence using spaCy[52] embeddings learned using a multi-task convolutional neural network.

For evaluating the compressions against the reference compressions, we use a Python wrapper[53] of the ROUGE metric Lin (2004) with the parameters suggested by Owczarzak et al. (2012) yielding high correlation with human judgments (i.e., with stemming and without stopword removal).[54] To evaluate and assess the effectiveness of our active learning-based sampling approaches, we set up our interactive text compression approach for the two state-of-the-art Seq2Seq models consisting of a generative model (Seq2Seq-gen) and a generate-and-copy model (Pointer-gen) as described in Section 5.2.1. For the neural Seq2Seq text compression experiments, we set the beam size and batch size to 10 and 30 respectively. We use the Adam optimizer (Kingma and Ba, 2015) for the gradient-based optimization. Finally, the parameters for the neural network parameters like weights and biases are randomly initialized.

In order to asses the effectiveness of AL for neural text compression we extend the OpenNMT[55] implementations with our interactive framework following Algorithm 3. The sampling strategy selects instances to be annotated interactively by the user in batches. Next, the neural text compression model is incrementally updated with the selected samples.

For the interactive setup, we simulate the users by by using the compression pairs from our corpus as the sentences annotated by the user. This allows us to conduct repeatable experiments and adjust our parameters in a controlled setting.

[51] https://github.com/google-research-datasets/sentence-compression
[52] https://spacy.io/
[53] https://github.com/pltrdy/files2rouge
[54] -n 2 -c 95 -r 1000 -a -m
[55] https://github.com/OpenNMT/OpenNMT-py

5.5 Quantitative Analysis

To better understand the effectiveness of AL-based interactive compression framework, we performed a quantitative analysis across two factors: (a) in-domain training, to identify active learning strategies to be used with a minimum of labeled instances, (b) domain adaptation, to identify instances to be annotated by the user for quicker model adaptation.

5.5.1 Analysis of In-domain Active Learning

For in-domain active learning experiments, we choose the Google News text compression training corpus and sample for corpus sizes between 10% and 100% in ten percent point steps. As a baseline, we use a random sampling strategy to test the state-of-the-art Seq2Seq neural text compression models. Figure 5.3 suggests that our coverage-based sampling (Coverage-AL) and diversity-based sampling (Diversity-AL) strategies outperform the random sampling strategy throughout all training sizes. A key observation is that our sampling strategies are behind the upper bound by just 0.5% ROUGE-2 when only 20% of the training data is used. Table 5.1 illustrates the results of our sampling strategies when 20% of the data is used for training. All the results are in comparison to the upper bound (UB) receiving 100% of the training data.

Methods	UB			Random			Coverage-AL			Diversity-AL		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
Seq2Seq-gen	59.94	52.08	59.78	61.60	50.03	61.37	**62.89**	**51.38**	**62.56**	62.54	50.19	62.13
Pointer-gen	79.26	71.77	79.08	71.61	61.15	71.28	**78.11**	**70.50**	**77.89**	77.45	70.30	77.38

Table 5.1: ROUGE-1 (R1), ROUGE-2 (R2) and ROUGE-L (RL) achieved by the state-of-the-art models using our sampling strategies evaluated on the Google compression test set. Bold marks best AL strategy.

Coverage-AL performs better than the Diversity-AL for both the Seq2Seq-gen and Pointer-gen models. However, they are still not effective in the Seq2Seq-gen model where random sampling performs on par with the active learning sampling approaches. We believe this is due to the Seq2Seq-gen model's inability to copy from the source text in the sampled set as a consequence of active learning in the batch setting. Whereas for Pointer-gen model, we observed that both Coverage-AL and Diversity-AL strategies of adding new samples for training had a greater impact when the model has not adapted. We attribute the effectiveness of the Coverage-AL strategy over Diversity-AL to the exploitation of the model uncertainty, as the Diversity-AL only uses the similarity based on the samples, but misses to integrate the model uncertainty.

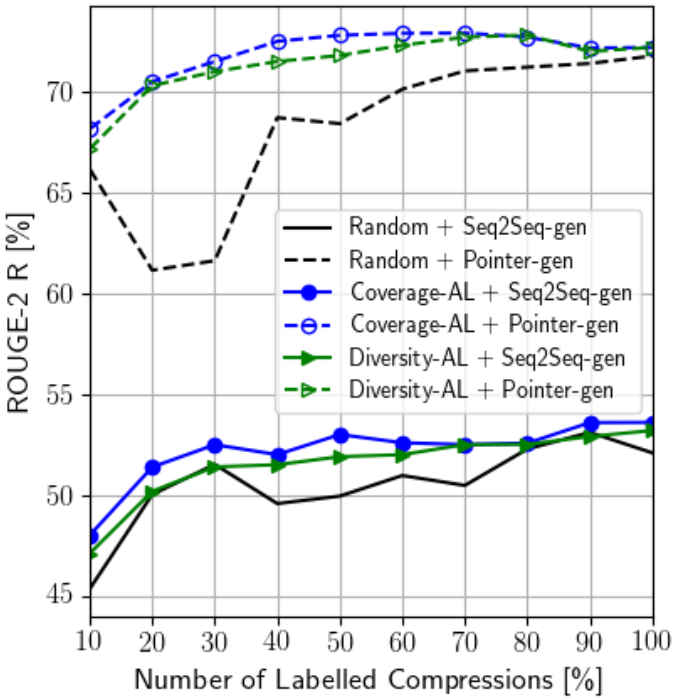


Figure 5.3: Analysis of the active learning approaches combined with state-of-the-art Seq2Seq compression models on Google compression dataset while varying the training sizes.

Table 5.2 presents an example sentence compression pair from the Google News dataset and the generated compressions of both neural Seq2Seq models when using one of the three sampling strategies. The example shows that detailed descriptions like the names of the ships "JING GANGSHA" and "HENG SHUI" are dropped by all models. In particular, the Seq2Seq-gen model has the problem of generating words not present in the original text (e.g., "toddlers", "Scottsbluff"). In contrast, the Pointer-gen model's ability to copy from the original text restrains the model from generating irrelevant words. Although Diverysity-AL based models recognized the phrasal constructs crucial for the sentence meaning, Coverage-AL generated the closest compression to the reference.

5.5.2 Analysis of Active learning for domain adaptation

To test our interactive Seq2Seq model using active learning strategies for the domain adaptation scenario, we train the model on the Google News compression corpus and test it on the multi-genre MSR-OANC compression dataset. Additionally, for domain adaptation, the neural Seq2Seq model is updated incrementally using our interactive compression Algorithm 3. The sampling strategies select the instances to be interactively annotated by the user. The two sampling strategies used for in-domain active learning are used for interactive compression with the state-of-the-art Seq2Seq models. Table 5.3 illustrates the results of the interactive text

Source text:	Two Chinese war ships , “ JING GANGSHA ” and “ HENG SHUI ” arrived at the port of Trincomalee on 13 th January 2014 on a good will visit .
Reference:	Two Chinese war ships , arrived at the port of Trincomalee will visit .
Seq2Seq-gen	
+ Random:	Two Chinese war ships , arrived at the port of toddlers on 13 th January 2014 .
+ Coverage-AL:	Two Chinese war ships , arrived at the port of Trincomalee on a good will visit .
+ Diversity-AL:	Two Chinese war ships arrived at the port of Scottsbluff on 13 th .
Pointer-gen	
+ Random:	Two Chinese war ships , arrived at the port of Trincomalee on 13 th January 2014 .
+ Coverage-AL:	Two Chinese war ships arrived at the port of Trincomalee will visit .
+ Diversity-AL:	Two Chinese war ships , arrived at the port of Trincomalee .

Table 5.2: In-domain active learning example sentence and compressions for Google News compression dataset when using 20% of labelled compressions with Random, Coverage-AL, Diversity-AL sampling strategies

Methods	MSR-OANC ID			Random			Coverage-AL			Diversity-AL		
	R1	R2	RL	R1	R2	RL	R1	R2	RL	R1	R2	RL
Seq2Seq-gen	30.05	10.42	26.87	33.51	13.60	30.26	**35.10**	**15.00**	**32.78**	34.85	14.92	32.41
Pointer-gen	35.24	16.57	32.56	38.19	21.87	37.94	**39.59**	**24.87**	**37.02**	39.42	24.70	36.86

Table 5.3: ROUGE-1 (R1), ROUGE-2 (R2) and ROUGE-L (RL) achieved by the state-of-the-art models using our sampling strategies when interactively retrained using 10% of the MSR-OANC training set. The results are in comparison to the models trained on in-domain training set (MSR-OANC ID). Bold marks best AL strategy.

compression model when applied to the MSR-OANC text compression dataset. One interesting observation is the fact that our sampling strategies with only 10% of the training data ($\approx$ 500 samples) perform better than models trained on in-domain training data (MSR-OANC ID) with 5k training instances by +8.3% and +8.2% ROUGE-2.

Figure 5.4 shows the results for the various sample sizes of the 5k training instances. The results show a similar trend as the active learning for the interactive data-selection scenario. The Coverage-AL and Diversity-AL strategies do not show significant differences from each other. However, the two active learning strategies achieve on average +2.5% ROUGE-2 better results than the random sampling. The results demonstrate that the use of relevant training samples is useful for transferring the models to new domains and genres. Another observation as compared to the results by Toutanova et al. (2016) is that training on a large Google News

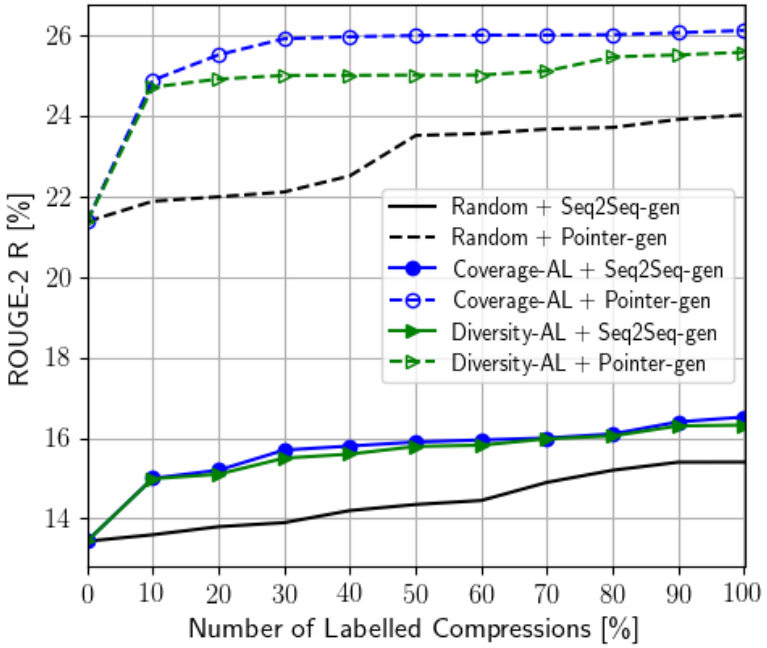


Figure 5.4: Analysis of the active learning for domain adaptation on the MSR-OANC dataset while varying the training data.

Source text:	Given the urgency of the situation in Alaska , Defenders needs your immediate assistance to help save Alaska 's wolves from same - day airborne land - and - shoot slaughter .
Reference:	Given the urgency of the situation in Alaska , Defenders needs your immediate assistance saving Alaska 's wolves from slaughter .
Seq2Seq-gen	
+ Random:	Immediate assistance to save Alaska's tundra .
+ Coverage-AL:	Sometimes needs your assistance to help save Alaska 's wolves .
+ Diversity-AL:	The situation in Alaska, help save Alaska 's tundra .
Pointer-gen	
+ Random:	Immediate assistance to help save Alaska 's wolves .
+ Coverage-AL:	The urgency of the situation in Alaska , Defenders needs your immediate assistance .
+ Diversity-AL:	Defenders needs your assistance to help save Alaska 's wolves .

Table 5.4: Domain adaptation example from the MSR-OANC dataset when trained on a 20% of labelled compressions with Random, Coverage-AL, and Diversity-AL sampling strategies

compression dataset and fine-tuning on the MSR-OANC compression dataset performs better than solely training on the small dataset.

Table 5.4 shows an example from the MSR-OANC compression dataset. The example illustrates similar compression properties as seen in the in-domain settings. In particular, the two models learned to drop appositions, optional modifiers, detailed clauses, etc. Additionally, we also observed that the difficult cases where those where there is little to be removed, but due to higher compression ratios during the training, the models removed more than required. This confirms the cause for lower ROUGE scores compared to the Google News corpus.

Lastly, to be fair between all the compared models, we also checked the compression ratios of all the compressed text outputs in the test dataset. All our models have similar compression ratios of $\approx$ 65% i.e., retain $\approx$ 65% of the source text. Also, since we force the summaries to compress at least 75% of the text, the fixed constraint makes the ROUGE-2 Recall measure appropriate as it is the ratio of the overlapping n-grams with the total number of n-grams in the reference summary. The short summaries will automatically be penalized as their overlap is smaller. However, a ROUGE-2 Precision will not be able to capture this penalization.

5.6 Chapter Summary

In this chapter, we focused on data-efficient training for neural text compression. Our contributions include a novel neural text compression approach using a neural Seq2Seq method with an interactive setup that aims at (a) learning an in-domain model with a minimum of data and (b) adapting a pretrained model with few user inputs to a new domain or genre.

For interactive text compression, we investigate two uncertainty-based active learning strategies with (a) a coverage constraint using attention dispersion and (b) a diversity constraint using structural similarity to make better use of the human in the loop for preparing training data pairs. The active learning based data selection methodology samples the data such that the most uncertain samples are available for training first. Experimental results show that the selected samples achieve comparable performance to the state-of-the-art systems, but trained on 80% less in-domain training data. Active learning with an interactive text compression model helps in transferring models trained on a large parallel corpus for a headline generation task to a general compression dataset with just 500 sampled instances. Additionally, the same in-domain active learning based data selection shows a notable performance improvement in an online interactive domain adaptation setup. Our experiments demonstrate that instead of more training data, relevant training data is essential for training Seq2Seq models in both in-domain training as well as domain adaptation.

Overall, switching from non-interactive to an interactive setup based on active learning strategies has greatly minimized the amount of necessary training data for neural text compression. This is an interesting finding that can potentially also alleviate the need for huge datasets in other other natural language processing tasks, such as question answering, to transfer a model to a new domain or genre.

Chapter 6

Information Recommendation

In this chapter, we focus on explainable models for information recommendation. We specifically focus on item rating prediction and review summarization as a scenario for journalism. In Section 6.1, we discuss the need for models to exploit information from reviews for building better user and item profiles and explain a model's recommendation at the same time. Then, we propose our novel joint rating prediction and review summarization approach combining explicit topic vectors in Section 6.3. In Section 6.4, we evaluate our approaches experimentally using standard recommendation datasets and analyze the performance of the approaches on rating prediction and review summary generation. In Section 6.5.3, we propose two ways to explain the predicted rating. Finally, we conclude with the summary of our findings in Section 6.6.

6.1 Motivation and Challenges

Item recommender systems have increasingly gained attention in the information retrieval and natural language processing communities, both in academia and industry. These systems help users digest vast amounts of information by adapting it to individual user interests, e.g., when deciding to see a movie or series on Amazon or selecting a sushi restaurant on Yelp. In journalism, news agencies use recommendation systems to reduce the clutter of news articles and present it to their readers. News media houses use the user information gathered through interactions, such as, clicks, user comments to recommend articles to the user. The recommendations are not limited to news articles, recommending user comments for author response, detecting fake news articles and toxic comment detection are a few applications. Most existing recommendation methods are based on collaborative filtering (Salakhutdinov and Mnih, 2007; Lee and Seung, 2000; Koren, 2008), which primarily learn users' and items' latent factors from ratings. Most past systems rely on user-provided ratings (e.g., 1–5 stars) to learn preference models, but such an approach fails to capture valuable information from

Users	Aspect Words
T.	Cast: fit, work Genre: action
hairbear2012	Genre: Spy, action Screnplay: storytelling
scooby	Screenplay: car chase Cost: bucks
Alex	Screenplay: chase scenes, human interaction

Figure 6.1: Example ratings, reviews and their summaries for Jason Bourne (2016) on Amazon Movies. Reviews describe detailed personalized opinion and interests of the user w.r.t. the item. The table on the right-hand-side shows extracted aspect words from the reviews modeling the users' preferences.

actual user experiences, which can be recorded in the form of reviews. This user-generated content is an increasingly important source, useful for both content providers as well as the end user. In this chapter, we propose *J3R*, a novel multi-task learning setup for explainable recommendation based on ratings *and* reviews, which we motivate below.

User and item profiles for recommendation. Although recommender systems based on reviews have been previously proposed, (Esparza et al., 2010; Musat et al., 2013; McAuley and Leskovec, 2013; Aciar et al., 2007), they yet do not fully exploit the potential of learning to recommend jointly from both reviews and ratings. Figure 6.1 shows four reviews on the Jason Bourne (2016) movie, which illustrate the connection between reviews and ratings: Each review consists of a brief summary (e.g., "Better with age" in T.'s review) and the actual review text in addition to the rating (i.e., 1–5 stars). The users focus on multiple different aspects in their reviews, including the main actor, cast, director, genre, screenplay, etc. For example, user T. likes Matt Damon's looks, fitness, and the action in the movie. In contrast, Alex and scooby have differing opinions on the use of car chases in the screenplay. The example shown is a typical real-world use case where different users have different interests and opinions about certain aspects of the same item. We aim at exploiting this information from reviews for building user and item profiles. Additionally, we leverage recent advances in deep neural networks to exploit the commonality between the rating and the review summary in a multi-task learning (MTL) approach where rating prediction is the main task and review summary generation is the auxiliary task.

Explainable Recommendation. In a recent review by Goodman and Flaxman (2017) on European Union regulations on algorithmic decision-making, the authors explain how the Article

22 of the European Union's new General Data Protection Regulation on automated individual decision-making, including profiling, potentially prohibits a wide range of algorithms currently in use, including recommendation systems, computational advertising, etc. The law effectively states "the right to explanation", where a user could ask for explanations on the decisions made by the algorithm about them. This regulation is only one recent development to strongly encourage the machine-learning-based communities to design algorithms in view of enabling explanations.

Although the primary goal of a recommender system is to produce better recommendations, it is a clear advantage if a system provides explanations for its users. Explanations serve multiple purposes, such as building trust, creating transparency, improving efficiency by quicker decision-making, and increasing user satisfaction (Tintarev and Masthoff, 2011). There has been a recent surge in methods focusing on explainable recommendation systems (Zhang et al., 2014; Catherine et al., 2017; Li et al., 2017b; Chen et al., 2018). Previous approaches use explicit topics from reviews with users' opinions (Zhang et al., 2014), knowledge graphs (Catherine et al., 2017), tip generation (Li et al., 2017b) and review ranking (Chen et al., 2018) for explanations.

In our research, we propose a novel approach to combine explicit topic vectors from reviews with generated review summaries as a way to explain a predicted rating. The final explanations of our *ʒ3R* system are thus of two types: (a) a histogram of user preferences on different topics of a domain, computed from the updated user vectors learned by our MTL approach and (b) a ten-word review summary of a review and the attention highlights on the review based on the weights learned from the user–item vectors. For the Jason Bourne example from Figure 6.1, a user vector for user T. should capture T.'s interest in the cast and the genre based on the user's past reviews. In addition to the histograms, based on the preferences from scooby's vector, the words in Alex's review would be highlighted according to their importance with respect to scooby's profile and the review would be automatically summarized. In Section 6.5.3, we discuss how our *ʒ3R* system implements this kind of explainable recommendation.

6.2 Related Work

In this section, we discuss recommendation systems as introduced in Section 2.4 specific to joint models of rating prediction and summary generation and explainable recommendation. Then, we introduce the state-of-the-art systems which we use as baselines.

In the field of recommendation, researchers have made attempts to combine different neural network architectures with collaborative filtering e.g., neural collaborative filtering (He et al., 2017), factorization machines (He and Chua, 2017), deep matrix factorization (Xue et al., 2017). In this work, we discuss joint models which use multi-task learning. Multi-task learning approaches have seen significant success in the area of machine learning and natural lan-

guage processing (Collobert and Weston, 2008; Rei, 2017; Liu et al., 2017). The goal of these approaches is to learn two related tasks which can mutually benefit from each other. As rating prediction and review summary generation are two facets of the same user preference of an item, they can be optimized together by sharing the parameters across the model. Although review summary generation has been conducted independently of rating predictions (Hu and Liu, 2004; Zhou et al., 2017; Ly et al., 2011; Wang and Ling, 2016), jointly modeling the rating prediction and the review summary generation has as yet only shown first promising results (Li et al., 2017b; Yu et al., 2016). In our work, we go beyond such models by employing pointer-generated neural models and an attention mechanism on user preferences which particularly benefit the auxiliary task of review summary generation.

Although state-of-the-art methods produce generally good recommendations, they fail to explain the reasons for a particular recommendation. Explanations can serve as a way to understand the algorithms and the models learned. This has led to new research questions for explaining recommendation systems and their output (Tintarev and Masthoff, 2011; Zhang et al., 2014; Catherine et al., 2017; Li et al., 2017b; Mukherjee et al., 2017; He et al., 2015; Ren et al., 2017). Some of the promising approaches include topic models as latent factors (He et al., 2015; Mukherjee et al., 2017; Zhang et al., 2014), knowledge graphs (Catherine et al., 2017), and tip generation (Li et al., 2017b; Li et al., 2019). Mukherjee et al. (2017) propose a joint model using reviews and ratings with a hidden Markov model and Latent Dirichlet allocation (LDA). They provide explanations with the help of words from latent word clusters explaining the essential aspects of the user and item pairs. Zhang et al. (2014) propose explicit factor models for generating explanations by extracting phrases and sentiments from user-written reviews for the items. In our approach, we combine multiple types of explanations and we generate them by jointly learning from reviews and ratings.

The work by Li et al. (2017b) first proposes a multi-task learning framework to predict ratings and generate abstractive review summaries, which they extended in Li et al. (2019) by proposing a personalized solution. A major difference between their task and ours is that we generate summaries from the reviews, whereas, they generate from user–item latent vectors and the review vocabulary. Thus, the summaries generated in their task tend to be overly general as discussed in their paper. On the contrary, in our work, our goal is not only to generate summaries but also to use summarization as a method to explain the important content in the reviews based on the user preferences. We leverage recent machine learning advances in pointer-generated networks (Vinyals et al., 2015; Gu et al., 2016; See et al., 2017) and attention-based mechanisms (Bahdanau et al., 2015) which support the accurate generation of summaries by attending on latent user–item vectors, the users' ratings, and their reviews. Besides, the user and item profiles capture user–item preferences based on the review text which are missing in the generic latent factors.

We now introduce approaches which we use as the baseline in the remaining chapter:

PMF: Salakhutdinov and Mnih (2007) propose Probabilistic Matrix Factorization, which is a Matrix Factorization method using Gaussian distribution to model the users and items latent factors. The aim of PMF is to find a factorization for the preference matrix consisting of user–item ratings by minimizing the root mean squared error. This model scales linearly with the number of observations and showed significant performances on large and sparse dataset.

NMF: Lee and Seung (2000) propose Non-negative matrix factorization model, which is a matrix decomposition technique which splits the user–item preference matrix into the product of a user matrix and item matrix. NMF works in cases were the data is very sparse, as the missing-values and non-negative elements assumption is inbuilt in the algorithm.

SVD++: SVD++ is an extension of Singular Value Decomposition proposed by Koren (2008). SVD is a matrix factorization method which creates the two low-matrices of user and item latent factors considering implicit feedback information. SVD++ takes into account the user and item biases.

HFT: McAuley and Leskovec (2013) propose Hidden Factors as Topics, which is a state-of-the-art method that combines latent rating dimensions with latent review topics using exponential transformation function to link the stochastic distributions.

DeepCoNN: Zheng et al. (2017) propose Deep Cooperative Neural Networks, which is a state-of-the-art method that jointly models users and items from textual reviews using two parallel neural networks coupled using a shared output layer.

DeepCoNN++: Extended version of Zheng et al. (2017), where the shared layer with Factorization Machine estimator is replaced with a neural prediction layer.

NAARE: Chen et al. (2018) propose Neural Attentional Regression with Reviews-level Explanation mode, which uses a similar two neural network architecture as DeepCoNN++. Additionally, they use a attention-based review pooling mechanism to select the reviews for modeling.

6.3 Joint Explainable Recommendation

To address the research problem, we divide the problem into three steps as shown in Figure 6.2: (1) First, we build *user and item models* to identify interpretable topic vectors of an item capturing different aspects of the item that users are interested in. (2) Then, we train a *rating prediction model* using these user and item models. (3) Finally, we generate review summaries to explain the recommendations of our system by jointly modeling *rating prediction* and *review summary generation* using an MTL approach of multi-layer perceptron and pointer-generated

networks that utilize the user and item models. Our final method is called *J3R* ('Joint MTL of Ratings and Review Summaries for Explainable Recommendation') and consists of these three components: user and topic models, rating prediction, and review summary generation. We introduce the individual components in the following subsections.

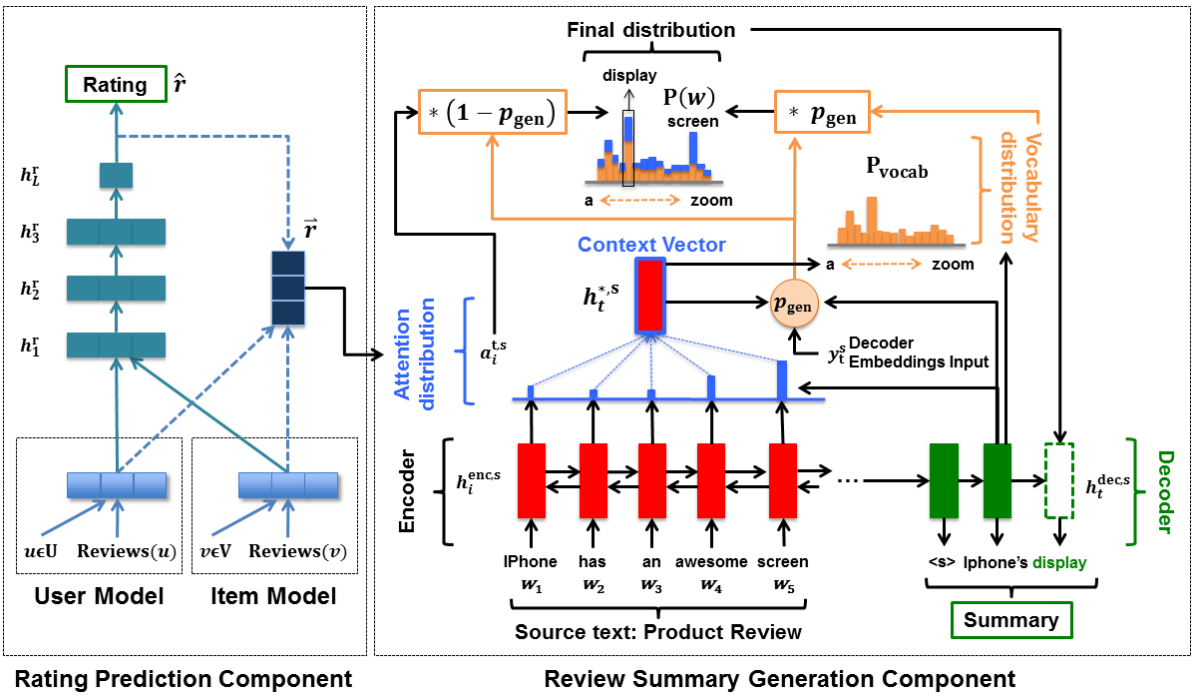


Figure 6.2: Model architecture of the aspect-based joint model for rating prediction and review summarization (*J3R*). The architecture is divided into three steps: (1) User and Item models; (2) Rating prediction model; (3) Review summarization model.

6.3.1 User and Item Models Component

The goal of this component is to build user and item profiles using the review content. To achieve this goal we first preprocess the data to identify all nouns and noun phrases from reviews (e.g., 'display', 'battery for a phone') of an item similar to Liu (2010). We collect all the nouns in a review as a bag-of-words representation to generate a 1,000-dimensional tf-idf vector, which captures the most frequent nouns describing a item in a domain.

These fixed-size tf-idf vectors are used as an input for the LDA (Blei et al., 2001) topic model to calculate the topic vectors. LDA is a probabilistic topic model which aims at finding a structure in the unlabeled text collection by identifying different topics based on the word usage. The probability distribution over high probability words gives us an understanding of the contents of the corpus. Thus, reviews grouped into different clusters using LDA can be

viewed as random mixtures over latent vectors, where a distribution over the most frequent nouns represents each topic.

Let D be a corpus of M reviews $D_1, D_2, \ldots, D_M$, where each review $D_i = (w_1, w_2, \ldots, w_N)$ is a sequence of N words from a vocabulary $\mathcal{W}$ and k the number of topics. Using LDA, we represent each document D_i as a k-dimensional topic distribution θ_d. Each topic vector, in turn, is an N-dimensional word distribution ϕ_k, which follows a Dirichlet prior β.

There are three steps to LDA: (1) it first draws a k-dimensional topic mixing distribution $\theta_d \sim Dir(\alpha)$ to generate a document d; (2) for each token w_{dn}, it draws a topic assignment z_{dn} from a multinomial distribution $Mult(\phi_{z_{dn}})$; and (3) finally, it draws a word $w_{dn} \in \mathcal{W}$ from $Mult(\phi_{z_{dn}})$ by selecting a topic z_{dn}.

To infer these latent variables (ϕ and θ) and hyperparameters (α and β), we compute the probability of the observed corpus:

$$p(D|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta) \right) \mathrm{d}\theta_d$$

We first use all the reviews $Reviews_u$ written by a user u and all reviews $Reviews_v$ of an item v respectively and turn them into N-dimensional tf-idf vectors. To generate topic vector profiles, we input these tf-idf vectors to the learned LDA topic model. The profiles learned using the user and item model are the initial latent vectors u and v for the rating prediction model discussed in the next section and are illustrated in Figure 6.2 as *User Model* and *Item Model*.

6.3.2 Rating Prediction Component

Our rating prediction component is illustrated on the left-hand side of Figure 6.2. It uses a traditional recommendation setup where the goal of the recommender is to predict the rating of a given user and item pair. We use a regression function to predict a rating score $\hat{r}$ based on the latent vector representations u and v of the users and items. Typical matrix factorization (MF) approaches do a linear transformation of these vectors as described in Eq. 6.1, where b is the global bias.

$$\hat{r} = u^T v + b \tag{6.1}$$

Although these linear transformations achieve state-of-the-art performance in recommendation systems, they cannot capture non-linear interactions between the users' and items' latent factors. Thus, we transfer knowledge from successful non-linear deep learning methods used in natural language processing for our task.

This can be achieved by concatenating the input vectors u and v as in Eq. 6.2:

$$h_1^r = relu(W_{h_1}^{\mathrm{r}}(u \oplus v) + b_{h_1}^{\mathrm{r}}) \tag{6.2}$$

where $W^{\mathrm{r}}_{h_1}$ is the weight matrix of the first hidden layer for the concatenated vector $u \oplus v$, the user's latent factors u, and the item's latent factors v, $b^{\mathrm{r}}_{h_1}$ is the bias term and $relu(x) = x^+ = max(0, x)$ the non-linear function. The superscript $^{\mathrm{r}}$ represents the parameters and variables for the rating prediction component of our model.

To further add non-linearity, we add layers of non-linear transformations:

$$h^{\mathrm{r}}_l = relu(W^{\mathrm{r}}_{h_l} h^{\mathrm{r}}_{l-1} + b^{\mathrm{r}}_{h_l}) \tag{6.3}$$

where l is the index of the hidden layers and $W^{\mathrm{r}}_{h_l}$ is the weight matrix. The number of hidden layers is a hyperparameter of our model.

Equation 6.4 describes the output layer with the weight matrix $W^r_{h_L}$. We use a sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$ to output a rating in the range [0,1], which we denormalize to the rating range (e.g., 1–5 stars) during the evaluation.

$$\hat{r} = \sigma(W^{\mathrm{r}}_{h_L} h^{\mathrm{r}}_L + b^{\mathrm{r}}_{h_L}) \tag{6.4}$$

To optimize the parameters and the latent factors u and v, we define the loss function:

$$\mathcal{L}^{\mathrm{r}} = \frac{1}{\mid \mathcal{X} \mid} \sum_{u \in \mathcal{U}, v \in \mathcal{V}} (\hat{r}_{u,v} - r_{u,v})^2 \tag{6.5}$$

where $\mathcal{X}$ is the training set, $\hat{r}_{u,v}$ is the predicted rating and $r_{u,v}$ is the gold-standard rating assigned by user $u \in \mathcal{U}$ to item $v \in \mathcal{V}$ respectively.

6.3.3 Review Summarization Generation Component

The goal of *J3R* is to mutually benefit from the available ratings and reviews in two different tasks: (a) rating prediction and (b) review summary generation. Rating prediction precisely aims at predicting the score for a given user and item pair, whereas the review summary generation component summarizes the review content using a sequence-to-sequence model based on user preferences. The user–item preferences (i.e. the user and item vectors) are shared with the rating prediction component, which are jointly learned using an MTL approach.

Our model is inspired by pointer-generated networks (Vinyals et al., 2015; See et al., 2017) to efficiently summarize the review, by using soft switching between copying words via pointing to the source text and generating words via a fixed vocabulary in a given context. The context in our generation setup consists of the user and item latent vectors $u \in \mathcal{U}$, $v \in \mathcal{V}$, the rating vector $\vec{r}$ (e.g. if the rating range is [1,5] then a rating vector for 3 stars is (0, 0, 1, 0, 0)), and the review D. The tokens of the review text $w_i \in D$ are provided as the input to the encoder one-by-one to produce a sequence of encoder hidden states $h^{\mathrm{enc,s}}_i$. At each time step t,

the decoder has the decoder states $h_t^{\text{dec,s}}$ which receives the word embeddings of the previous word as the input.

An important characteristic of our architecture is the attention distribution $a_i^{\text{t,s}}$ that we compute at each time step t with the encoder states $h_i^{\text{enc,s}}$, the decoder state $h_t^{\text{dec,s}}$, the user vector u, the item vector v, and the rating vector $\vec{r}$ as shown in Eq. 6.6–6.8. It can be viewed as a probability distribution over the source words, user preferences, item factors and rating, which tells the decoder which word to generate.

$$e_i^{\text{t,s}} = q^T tanh(W_h^{\text{enc,s}} h_i^{\text{enc,s}} + W_h^{\text{dec,s}} h_t^{\text{dec,s}} + W_r^{\text{s}}(u \oplus v \oplus \vec{r}) + b_{\text{att}}^{\text{s}}) \tag{6.6}$$

$$a_i^{\text{t,s}} = \frac{\exp(e_i^{\text{t,s}})}{\sum_{i'=1}^{N} \exp(e_{i'}^{\text{t,s}})} \tag{6.7}$$

where q, $W_h^{\text{enc,s}}$, $W_h^{\text{dec,s}}$, W_r^{s} and $b_{\text{att}}^{\text{s}}$ are learnable parameters and N is the number of words in the review text. The superscript $^{\text{s}}$ represents the parameters and variables for the review summary generation component of our model.

Using the attention distribution $a_i^{\text{t,s}}$, we compute the weighted sum of the encoder hidden states, also known as the context vector $h_t^{*,\text{s}}$ as shown in Eq. 6.8.

$$h_t^{*,s} = \sum_i a_i^{\text{t,s}} h_i^{\text{enc,s}} \tag{6.8}$$

To get the vocabulary distribution P_{vocab} at time step t, we concatenate the context vector with the decoder state $h_t^{\text{dec,s}}$ and pass it through two linear layers:

$$P_{\text{vocab}} = softmax(Q(Q' h_t^{\text{dec,s}} \oplus h_t^{*,\text{s}} + b'^{\text{s}}) + b^{\text{s}}) \tag{6.9}$$

where Q, Q', b^{s} and b'^{s} are learnable parameters.

To finally generate words, we use a pointer-generated network which decides whether to generate the word from the vocabulary P_{vocab} or copy one from the input sequence by sampling from the attention distribution $a^{\text{t,s}}$ as shown in Eq. 6.11. This is done by calculating an additional generation probability $p_{\text{gen}}^{\text{s}}$ for time step t, which is calculated from the context vector $h_t^{*,\text{s}}$, the decoder state $h_t^{\text{dec,s}}$, and the current input to the decoder y_t^{s}:

$$p_{\text{gen}} = \sigma(W_{h^*}^T h_t^{*,\text{s}} + W_{h^{\text{dec}}}^T h_t^{\text{dec,s}} + W_y^T y_t^{\text{s}} + b_{\text{gen}}^{\text{s}}) \tag{6.10}$$

$$P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i=0}^{N} a_i^{\text{t,s}} \tag{6.11}$$

where W_{h^*}, $W_{h^{\text{dec}}}$, W_y, $b^{\text{s}}_{\text{gen}}$ are learnable parameters and N is the number of words in the source review. Pointer-generator networks are helpful for handling out-of-vocabulary (OOV) words: if w is an OOV word then $P_{\text{vocab}} = 0$ and the word from the source review text is considered for generation and vice versa.

Finally, we define the loss function for the review summary generation component for the whole sequence as the normalized sum of the negative log likelihood of the target words w_t^*:

$$\mathcal{L}^s = -\frac{1}{T}\sum_{t=0}^{T} \log P(w_t^*) \tag{6.12}$$

6.3.4 Multi-task Learning Setup

We use a multi-task learning setup to jointly optimize the rating prediction and the review summary generation components by using a joint loss function $\mathcal{L}^j$:

$$\mathcal{L}^j = \lambda_r \mathcal{L}^r + \lambda_s \mathcal{L}^s + \lambda_o(||\mathcal{U}||_2^2 + ||\mathcal{V}||_2^2 + ||\Omega||_2^2) \tag{6.13}$$

where $\mathcal{L}^r$ is the rating regression loss from Equation 6.5 and $\mathcal{L}^s$ is the review summary generation loss from Equation 6.12. For regularization, we use the L2 norm $||\cdot||_2$ on the set of neural network parameters Ω, the user latent factors $\mathcal{U}$ and the item latent factors $\mathcal{V}$. λ_r, λ_s, λ_o are learnable parameters.

6.4 Evaluation Setup

6.4.1 Datasets

For our experiments, we use the Amazon 5-core[56] dataset on CDs, Toys, Music, Kindle, Electronics, Movies&TV and the Yelp 2018 dataset which are common benchmarks for recommendation systems introduced in Section 3.1.3. To pre-process the datasets, we perform tokenization, part-of-speech tagging and stemming with NLTK[57]. For the summary generation, we represent words using the Google News embeddings for English. Table 6.2 presents detail statistics of each dataset in terms of the number of reviews, users, items and vocabulary size.

We divide each of the datasets into training, development and testing consisting of 80%, 10% and 10% of the data respectively, which is a typical split ratio in recommendation evaluation. For all baseline methods (PMF, NMF, SVD++, HFT[58]), we used the Librec toolkit[59] and we

[56] http://jmcauley.ucsd.edu/data/amazon
[57] https://www.nltk.org/
[58] https://github.com/lipiji/HFT
[59] https://www.librec.net/dokuwiki/doku.php?id=Recommender

Dataset	Reviews	Users	Items	User Vocab	Item Vocab
CDs	1,097,592	75,258	64,443	363,883	418,414
Toys	167,597	19,412	11,924	56,456	59,414
Music	64,706	5,541	3,568	78,293	83,904
Kindle	982,619	68,223	61,934	184,885	205,915
Electronics	1,685,748	192,403	63,001	256,920	235,408
Movies	1,697,533	123,960	50,052	397,060	495,021
Yelp	3,072,057	199,445	115,798	335,831	340,526

Table 6.2: Statistics of the number of Reviews, Users, Items and Vocabulary of Amazon (CDs, Toys, Music, Kindle, Electronics, Movies) and Yelp Datasets

selected the number of latent factors for each domain after fine tuning on the development set. The number of factors for CDs, Toys, Music, Kindle, Electronics, Movies and Yelp dataset are 30, 30, 10, 20, 10, 10 and 100 respectively. To calculate the topic vectors, we set the tf-idf vectors size to 1,000. For our neural network based approach, after hyperparameter fine tuning, we set the latent factors to 32 and the number of hidden layers to 2. For the gradient-based optimization, we use the Adam optimizer.

For review summary generation, we consider the pairs of (reviews, review summary) written by the user as (input text, gold standard summary). For initializing the parameters, we first set the beam size to 10, which is a parameter in the beam search algorithm that determines the number of best partial solutions to evaluate while decoding. For a summary generation, for example, beam size limits the number of candidates to consider during decoding. We also set the maximum summary length to 10, as nearly 80% of the summaries are less than or equal to 10 words. Finally, the neural network parameters, like weights and biases, are randomly initialized.

6.4.2 Evaluation Metrics

To evaluate the rating prediction component, we employ two widely used metrics for recommender systems: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

$$MAE = \sum_{u,v} \frac{|r_{u,v} - \hat{r}_{u,v}|}{n} \tag{6.14}$$

$$RMSE = \sqrt{\sum_{u,v} \left(\frac{r_{u,v} - \hat{r}_{u,v}}{n}\right)^2} \tag{6.15}$$

Where, $r_{u,v}$ is the ground-truth rating, $\hat{r}_{u,v}$ is the predicted rating for a given user u and item v pair, n is the total number of ratings between users and items.

To evaluate the review summary generation component, we use ROUGE scores (Lin, 2004) between the generated summary and the gold standard summary. for each review in the test set. The ROUGE scores are calculated with the parameters suggested by Owczarzak et al. (2012) yielding high correlation with human judgments (i.e., with stemming and without stopword removal).[60]

6.5 Quantitative Qualitative Analysis

To better understand the performance of our recommendation system, we organize our quantitative and qualitative analysis across three factors: (a) rating prediction (b) summary generation, and (c) explainability.

6.5.1 Rating Prediction Analysis

Table 6.3 shows the results of the rating prediction component in comparison to our baselines. It shows that our model *J3R* consistently outperforms all other methods in terms of MAE and RMSE scores on all datasets. We also observe that the collaborative filtering methods PMF and NMF have low performance scores compared to other baselines. However, SVD++ shows that it is still a strong baseline for recommendation systems as shown in the Netflix Prize 2008.[61] SVD++ performs on par or better in comparison to the state-of-the-art neural content based systems like DeepCoNN, DeepCoNN++ and NARRE on small and medium sized data, however, the neural approaches perform better on large datasets. Overall, the results show that our *J3R* (Pointer) model performs better in terms of MAE and RMSE scores as compared to the best comparison methods NAARE and SVD++. This shows that review information helps in improving the representation of the user and item latent factors, which is further enhanced with the joint learning of rating prediction and review summary generation. The improvement is consistent and significant across the six datasets, whereas it is slightly lower on the Music dataset (-1.5%) compared to Electronics ($+2.9\%$), Movies&TV ($+2.2\%$) or Yelp ($+4.0\%$). The lower scores for Music is due to fewer reviews available for content-based models, which explains that latent factors of SVD++ also capture better information when there is less training data. Lastly, as the MAE and RMSE scores for the top 4 models are close, we significance of these scores using p-value, i.e., to check the probability that the obtained results were not due to luck. We compared our model *J3R* (Pointer) with NAARE, SVD++, DeepCoNN++ and observed that our algorithm is significantly better than the other models in all the datasets except for Music with 95% confidence ($p < 0.05$).

[60] -c 95 -r 1000 -n 2 -a -m

[61] https://www.netflixprize.com/community/topic_1537.html

Models	CDs		Toys		Music		Kindle		Electronics		Movies		Yelp	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Baselines														
PMF	0.682	0.972	0.705	0.979	0.849	0.922	0.573	0.835	0.855	1.193	0.765	1.083	0.967	1.273
NMF	0.749	1.082	0.693	0.999	0.700	0.997	0.651	0.956	0.952	1.366	0.830	1.176	1.024	1.381
SVD++	0.667	0.956	0.636	0.907	**0.641**	0.905	0.540	0.790	0.848	1.163	0.750	1.043	0.953	1.236
HFT	0.746	0.979	0.645	0.892	0.665	0.911	0.664	0.869	0.846	1.112	0.838	1.076	1.028	1.252
DeepCoNN	0.695	0.944	0.669	0.912	0.672	0.901	0.565	0.791	0.866	1.124	0.750	1.016	0.938	1.186
DeepCoNN++	0.682	0.933	0.652	0.900	0.659	0.894	0.553	0.783	0.824	1.113	0.742	1.002	0.922	1.202
NARRE	0.675	0.930	0.683	0.906	0.698	0.925	0.547	0.785	0.834	1.107	0.736	1.001	0.921	1.186
Our Models														
MLP	0.751	0.995	0.695	0.967	0.710	0.990	0.627	0.857	0.875	1.167	0.816	1.083	0.997	1.324
MLPTopic	0.706	0.954	0.674	0.943	0.685	0.907	0.602	0.814	0.839	1.113	0.758	1.059	0.967	1.258
J3R (Seq2Seq)	0.685	0.937	0.647	0.899	0.660	0.892	0.560	0.794	0.823	1.052	0.746	1.008	0.919	1.174
J3R (Pointer)	**0.661***	**0.912***	**0.634***	**0.880***	0.656	**0.890**	**0.538***	**0.775***	**0.805***	**0.995***	**0.714***	**0.984***	**0.881***	**1.009***

Table 6.3: MAE and RMSE scores for our models in comparison to the state-of-the-art models (lower is better). * denotes that *J3R* (Pointer) performs better than NAARE, SVD++ and DeepCoNN++ with statistical significance with $p < 0.05$. ***bold*** denotes the best performing models with least error.

Ablation Analysis. To quantify the impact of each component on the rating prediction task, we do an ablation analysis. We try two different settings contrasting two single-task learning setups with our MTL setup: (a) *MLP*: the rating prediction component (section 6.3.2), where a multi-layer perceptron based rating prediction model is randomly initialized with user and item vectors, (b) *MLPTopic*: the rating prediction component plus the topic vector component (section 6.3.1 and 6.3.2) and (c) *J3R*: the full setup of all three components introduced in Section 6.3, which uses a multi-task learning framework to jointly predict ratings and generate review summaries using user and item topic vectors initialized by the LDA topic vectors. *J3R* (Seq2Seq) is an alternative to Li et al. (2017b), where the GRU layers are replaced with LSTM and the rating regression has three hidden layers instead of one. *J3R* (Pointer) is our proposed method.

Table 6.3 shows that *MLPTopic* performs better than the simple *MLP* model, which explains that the LDA topic vectors are useful for rating prediction as they capture user–item preferences. Our best performing model *J3R* (Pointer) outperforms the individual components consistently across different domains. This elucidates that multi-task learning of rating prediction with review summary generation initialized with LDA based user and item models capture better user and item latent vectors. Furthermore, *J3R* (Pointer) performs better than *J3R* (Seq2Seq) and shows that the use of pointer network helps to improve the predictions.

6.5.2 Review Summary Generation Analysis

Although summarization is our auxiliary task to assist our main task of rating prediction, we separately evaluate the performance of our review summary generation component in

Models	CDs		Toys		Music		Kindle		Electronics		Movies		Yelp	
	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
TF·IDF	.078	.017	.097	.027	.079	.019	.087	.024	.098	.029	.087	.023	.191	.126
LexRank	.087	.021	.107	.031	**.087**	**.024**	.097	.024	.109	.035	.096	.027	.204	.126
LSA	.068	.012	.077	.018	.068	.013	.070	.015	.081	.020	.074	.016	.122	.061
KL-Greedy	.070	.013	.080	.018	.073	.015	.074	.017	.086	.023	.078	.017	.141	.079
ICSI	.047	.010	.064	.017	.043	.008	.058	.017	.061	.018	.050	.012	.119	.064
Seq2Seq-gen	.108	.025	.114	.026	.053	.005	.139	.035	.177	.065	.134	.040	.219	.131
Pointer-gen	.135	.039	.122	.030	.059	.007	.152	.047	.179	.069	.141	.052	.250	.163
J3R(Seq2Seq)	.119	.030	.120	.031	.060	.010	.150	.042	.185	.078	.145	.059	.235	.148
J3R(Pointer)	**.156**	**.045**	**.137**	**.040**	.065	.012	**.185**	**.053**	**.190**	**.082**	**.159**	**.065**	**.274**	**.181**

Table 6.4: ROUGE-1 (R1) and ROUGE-2 (R2) Precision scores of different summarization systems

this section. Table 6.4 shows the comparison of the review summary generation of *J3R* with baseline summarization models introduced in Section 4.2.

LexRank is the best-performing method among all the extractive baselines and performs the best on the Music dataset. However, the results show that the generative methods i.e. Seq2Seq-gen and Pointer-gen, improve in ROUGE-1 and ROUGE-2 when compared to the baseline systems on the other six datasets, whereas for the Music dataset the results are only slightly lower than the best performing system *LexRank*. Our model, *J3R* (Pointer) performs the best among all the generative methods, exhibiting that multi-task learning based method captures user importance during summary generation. For the Music domain, we observe that the generative methods perform worse than the extractive methods due to the small data size available for training. Another reason is that *J3R* (Pointer)'s pointer-generator networks tend to produce short abstractive summaries, while the extractive baselines produce longer summaries increasing the chances of overlaps with the gold summary. Furthermore, from the data analysis across datasets we observe that about 30% of the dataset have zero ROUGE-1 and ROUGE-2 scores, which explains the overall low ROUGE-1 and ROUGE-2 across various methods.

Lastly, it is important to note that although *J3R* (Pointer) combines different kinds of information efficiently (e.g., rating, review summary, item topics), one drawback of such a complex system is the difficulty in fine-tuning the system as there are a large number of parameters.

6.5.3 Explainability Analysis

Besides performance improvements, an important advantage of our *J3R* system is the interpretability of the generated recommendations. In this section, we analyze two ways of explanations: (a) illustrating the importance of different topics with respect to a user based on

Figure 6.3: Interpretation of the user preferences using an histogram over top five topics from the topic model (left) . Word importance on the source review shows the evidence for the predicted rating (right).

Director	Genre	DVD	Cast	Cinema	Food	Service	Cuisine	Breakfast	Price
seasons	story	video	actor	scene	restaurant	server	greek	egg	check
episodes	style	collection	role	family	main course	menu	chinese	eat	pay
part	horror	quality	performance	love	taste	time	pizza	pancake	money
point	comedy	television	voice	relationship	experience	owner	rice	sandwich	stay
release	drama	series	dialogue	experience	soup	stay	ramen	fresh	cost

Table 6.5: Top five words for each of the top five topics of Movie&TV (left) and Yelp restaurant domain (right) explained with the most representative words.

topic vectors and (b) illustrating the word importance in the reviews while summarizing the content for the user.

First, our user model described in Section 6.3.1 illustrates the user's preferences on the important aspects of a domain. Table 6.5 shows the top five topics with their most representative word and the top five words describing each topic in the Movies&TV and the Yelp restaurant domain. The topic words i.e., the most representative word, has been picked manually from the top two words for each topic. To gain a better interpretation of the topic words, we remove words belonging to multiple topics. Thus, based on the topic distribution θ_d of important words in a domain and the distribution of the words $\phi_{z_{dn}}$ across a topic, a user's preferences are computed from the user vector u created from the reviews written by the user. An example explanation of the preferences of a user who has written 490 reviews in Movies&TV is shown in the histogram on the left-hand side of Figure 6.3.

Second, we use the representative words in a review as evidence to explain the rating. We investigate word importance in the review using the attention weights. Figure 6.3 illustrates an example from the Movie&TV domain on Jason Bourne (2004). In the figure, we describe a scenario where the user decides to buy the DVD of Jason Bourne (2004). The user is overwhelmed by hundreds of reviews before making up the mind about the movie. Our *J3R* model summarizes each review and illustrates the most representative words of the review using

the attention mechanism as described in Section 6.3.3. On the right-hand side of figure 6.3, we highlight the word importance in the source review based on attention weights while generating a review summary. The example shows that phrases like "the spy thriller", "entertaining", "surpasses the original" are highlighted by our model for the generated summary "a good spy thriller". Furthermore, the generated summary and the gold standard summary illustrate the same aspects of a movie (e.g. "genre", "director style").

6.6 Chapter Summary

In this chapter, we propose explainable recommendation to keep human in the loop in the recommendation process. Our goal is not only to learn from the user, but also allow them to interpret what has been learned. To this end, we learn user preferences from ratings and reviews by using multi-task learning (MTL) of rating prediction and summarization of item reviews. Review summaries and ratings represent an overall user experience of a product. Although they are two separate entities, they capture similar information about aggregated user preferences. Reviews of an item tend to describe detailed user preferences (e.g., the cast, genre, or screenplay of a movie). A summary of such a review or a rating describes an overall user experience of the item. We propose a novel explainable recommendation system *J3R* using an MTL approach to jointly model user rating prediction and review summary generation. Our review summary generation model uses a pointer-generator network with an attention-based framework on user preferences and product attributes extracted from review content and user ratings as context vectors to generate summary text, which in turn is used as evidence for explaining the predicted rating. We empirically provide evidence for joint learning of rating prediction and summary generation being beneficial for recommendation by conducting experiments on the Yelp dataset and six different domains of the Amazon 5-core dataset. Additionally, we provide two ways of explanations (a) visualizing the user vectors on different topics of a domain, computed from our MTL approach and (b) a ten-word review summary of a review and the attention highlights generated on the review based on the user–item vectors.

Chapter 7

Conclusion

In this thesis, we focused on research question in the area of information preparation with human in the loop and presented example scenarios in journalism. In the following, we provide a summary of our findings and contributions and conclude with promising future directions for the research in the field of information preparation with human in the loop.

7.1 Summary and Contributions

In this thesis, we showed that information preparation in the areas of information summarization, condensation and recommendation alleviates the problem of information overload by putting the human in the loop. We therefore proposed a new live blog summarization task for information preparation in journalism. We proposed human in the loop approaches across the these areas i.e. interactive summarization system, where the human provides feedback on the concepts in the summary, interactive text compression, where the human steers the training process by active learning. Additionally, we achieved time and data efficiency techniques by approximating sampling and active learning. And finally, we proposed an explainable recommendation system which used a joint model of rating prediction and review summarization.

Chapter 2 We argue that information overload in journalism can be alleviated by information preparation which involves discovering important information, filtering redundant and unnecessary information, and adapting to the user's information need. Thus, divide the corresponding methods in information preparation into three areas: summarization, condensation, recommendation, for which discussed previous work. All the methods are dependent on user's needs, which are only insufficiently covered by existing solutions. We therefore proposed methods to put the human in the loop to address them in the areas of information summarization, condensation and recommendation.

Chapter 3 surveyed the datasets available in the areas of information summarization, condensation and recommendation. Additionally, the chapter introduced the new task of live blog summarization which has direct impact in digital journalism. To this end, we developed a live blog summarization dataset for two news websites namely, BBC and The Guardian, consisting of 762 and 1,683 topics respectively. Our live blog corpus is highly heterogeneous as compared to the standard DUC and TAC datasets. Further, it poses new challenges in the field of news summarization and we motivated the need for new methods in discussed in the remaining thesis. Lastly, the software for creating this corpus is publicly available and can be transferred to other sources of live blogs.

Chapter 4 focused on the task of multi-document summarization as a way of alleviating information overload by summarizing information. In this chapter, we addressed our first research question of investigating the performance of the state-of-the-art systems on information summarization. We showed that the previous works mainly focused on creating an optimized method to generate one single best summary that fits all users. Although these systems achieve state-of-the-art performance, we argue that these systems are highly impractical and of limited usefulness in the real-world application as they do not take the user's goal into account. To address these drawbacks and our second research question of integrating human in the loop, we proposed an interactive human-in-the-loop framework to create multi-document summaries that learn to adapt to the user's information need. In addition to that, we investigated sampling strategies based on active learning and joint optimization to reduce the number of iterations and the amount of user feedback. We identified five factors for quantitatively and qualitatively analyze our results, namely, effect of user feedback methods, concept notion, user study, and scalability. The results showed that AL model is best across datasets. Our interactive system took 10 iterations, confirmed in both simulation and user study. Additionally, by using a new approximate model, we can ensure computation times of less than 500ms which scales well for applications with real-time interactions. Our approach performed well on homogeneous data and on datasets covering multiple heterogeneous sources, which are typically harder to summarize. Additionally, our approach can be applied to multiple languages with less effort. Finally, we demonstrated two real-world use cases of how our interactive summarization system can be used for exploring large document collections. The interactive summarization system provides a generic summary in the first iteration and user iteratively interacts with it to satisfy user's information need. Besides the specific scenarios demonstrated in the chapter, the system can be used by journalists as a live blog summarization tool for automatically summarizing important information during a live event.

Chapter 5 introduced the task of text compression as a way to condense the abundant information. We addressed the problem of neural models being data hungry and proposed a new interactive text compression method to solve the need for large training data. To this end, we

employed state-of-the-art Seq2Seq text compression methods as our base models to address our first research question. Further, to address our next research question about data-efficient learning, we proposed an active learning-based human-in-the-loop setup with multiple sampling strategies to efficiently use minimal training data. We found that this new method of interactive text compression with intelligent sampling compressions from unlabeled corpora substantially reduces the amount of data needed to train the Seq2Seq model. As a second experiment, we further showed that that our interactive method can successfully transfer to a new domain with just a small amount of user input/annotations. Our final model was able to adapt from news headline generation to generic text compression with only 500 samples.

Chapter 6 focused on the task of information recommendation to address the information overload. In this chapter, we addressed the research question of how to use user preferences to have explainable information preparation. We mainly focused on explainable systems for which there is almost no previous work. We discuss the need for explainable recommendation models which devise interpretable models that work like a human and keep them in the loop during recommendation. We propose a new joint recommendation system of rating prediction and review summarization. In comparison to multiple baselines and state-of.the-art systems, we found our approach improves the rating prediction performance and also summarizes reviews based on the user preferences. About explaining the ratings, we generated a ten-word review summary marked with attention based on the user profiles, and a histogram of user preferences learned from the reviews of the users.

7.2 Future Research Directions

Following the research presented in this thesis, there are several opportunities for further research in information preparation with the human in the loop. As discussed in the individual chapters, the existing research in information preparation with human in the loop was limited, our work provides an excellent foundation for research in summarization, condensation and recommendation. During our research many possible future directions emerged such as different types of feedback, scalable approaches, types of sampling approaches, low-resource learning and cold-start settings. In view of the potential for future improvements, we point out a few promising directions:

Our interactive summarization method reach near the upper bound upper-bound in a few iterations. Our experiments primarily focused on point-based feedback on the concepts occurring in the summary. To leverage the potential of the interactive framework, it is worthwhile to explore other feedback types like preference-based (Zopf, 2018b) or sentence-based feedback (Saggion and Lapalme, 2002). Our work inspired the work by Gao et al. (2018), who propose APRIL, an interactive learning model which combines active preference learning and

reinforcement learning for summarization. Their model learns from preference-based user feedback on two summaries which is an alternative to our concept-based user feedback. Furthermore, APRIL uses noisy oracles (Viappiani and Boutilier, 2010) based on the user-response models, which is a good alternative to our perfect oracles and may yield more realistic simulation experiments. Other training methodologies like sample-efficient Bayesian-based active preference based algorithms (Simpson and Gurevych, 2018) can be combined with preference-based feedback to learn the prefernce-based interactive summarization model.

Furthermore, an alternative direction to approach interactive summarization is the summary representation. Other summary representations can be used to collect feedback such as structured summaries. Falke (2019) proposed a concept-map-based multi-document summarization as a variant to traditional text-based summarization. A similar procedure can be followed to showcase an optimal summary concept map as they also use an ILP-based backend. The interactive personalization of summary concept maps is therefore another interesting research direction for future work. Alternatively, Tauchmann et al. (2018) proposed hierarchical summaries for large text collections. An interactive hierarchical summarization framework with even more different types of feedback can be studied, such as, learning from how a graph is restructured or how a user navigates a graph structure in the summarization process.

Our data-efficient text compression framework in Chapter 5 showed that the application of active learning and interactive compression framework works and can be used in various applications built on Seq2Seq models. However, our results show that Seq2Seq models for low-resource settings are still far behind models trained on large datasets. Other training methodologies like incidental supervision (Roth, 2017) would probably better integrate the human in the loop in the process of information preparation. Such models could be indirectly supervised by providing feedback based on the behavior of the model. Other approaches like transfer learning based on fine-tuning language models (Howard and Ruder, 2018), or pre-training representations (Devlin et al., 2018), denoising auto-encoders (Févry and Phang, 2018) also seem to be promising directions to for low-resource settings as they have shown significant performance for many NLP tasks.

On a methodological level, there have been multiple new neural architectures proposed in natural language processing and machine learning that can yield improved performance and new ways to incorporate the users into the system's predictions. Latest architectures such as deep reinforcement learning frameworks (Wu and Hu, 2018), alternating pointer-networks (Jadhav and Rajan, 2018), hierarchical structured self-attentive network (Al-Sabahi et al., 2018) are some possible alternatives. Additionally, researchers have recently tried sampling by estimating intrinsic and externsic uncertainty in machine translation (Ott et al., 2018; Liu et al., 2018b) and self-training in text classification (Li et al., 2019). Similarly, new representations for input data like BERT (Devlin et al., 2018) and user feedback that better take the context into account would be useful to reduce the amount of feedback required. It would be interesting

to establish whether these additions improve further or degrade the performance.

Explainable Machine Learning is a budding field in not only in recommendation systems but also in NLP, ML and AI (Gurevych et al., 2017; Goebel et al., 2018). In the field of recommendation, textual explanation are garnering popularity as the user-generated content is abundant. In our research we have used topic models initialize our user and item representations, however, other representations like personalized embeddings (Li et al., 2019), attention-based context embeddings (Wang et al., 2018), social network based graph embeddings (Bourigault et al., 2014) are some alternatives. Additionally, systems still struggle to perform well in a cold-start setting (i.e., where users have not written sufficiently large number of reviews). It thus becomes hard to model user preferences in such cases. To address this problem alternative techniques like representative based learning (Liu et al., 2011) or contextual bandits (Auer et al., 2002), can be used in low-resource cold start setting. In low-resource settings relying on representatives of the set of items and users is a research direction for future work in explainable recommendation.

The discussion in this thesis shows that information overload is a serious challenge, not only in journalism but in many areas of our daily lives, including scientific research, business, and education. Our approach of information prep with the human in the loop shows promising results and we envision its use in other tasks where a computer and human work hand-in-hand.

List of Tables

List of Figures

Bibliography

Silvana Aciar, Debbie Zhang, Simeon J. Simoff, and John K. Debenham: 'Informed Recommender: Basing Recommendations on Consumer Product Reviews', *IEEE Intelligent Systems* 22 (3): 39–47, 2007.

Heike Adel, Benjamin Roth, and Hinrich Schütze: 'Comparing Convolutional Neural Networks to Traditional Models for Slot Filling', in: *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 828–838, 2016, Online: http://aclweb.org/anthology/N/N16/N16-1097.pdf.

Kamal Al-Sabahi, Zuping Zhang, and Mohammed Nadher: 'A Hierarchical Structured Self-Attentive Model for Extractive Document Summarization (HSSAS)', *IEEE Access* 6: 24205–24212, 2018, Online: https://doi.org/10.1109/ACCESS.2018.2829199.

Vicent Alabau, Christian Buck, Michael Carl, Francisco Casacuberta, Mercedes García-Martínez, Ulrich Germann, Jesús González-Rubio, Robin L. Hill, Philipp Koehn, Luis A. Leiva, Bartolomé Mesa-Lao, Daniel Ortiz-Martínez, Herve Saint-Amand, Germán Sanchis-Trilles, and Chara Tsoukala: 'CASMACAT: A Computer-assisted Translation Workbench', in: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pp. 25–28, 2014, Online: http://aclweb.org/anthology/E/E14/E14-2007.pdf.

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut: 'Text Summarization Techniques: A Brief Survey', *International Journal of Advanced Computer Science and Applications* 8 (10): 397–405, 2017.

David Allen and T. D. Wilson: 'Information overload: context and causes', *The New Review of Information Behaviour Research* 4 (1): 31–44, 2003.

Chinatsu Aone, Mary Ellen Okurowski, James Gorlinsky, and Bjornar Larsen: 'A trainable summarizer with knowledge acquired from robust NLP techniques', in Inderjeet Mani and

Mark T. Maybury (Eds.): *Advances in Automatic Text Summarization*, pp. 68–73, Cambridge, MA: MIT Press, 1995.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer: 'Finite-time Analysis of the Multiarmed Bandit Problem', *Machine Learning* 47 (2-3): 235–256, 2002, Online: https://doi.org/10.1023/A:1013689704352.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio: 'Neural Machine Translation by Jointly Learning to Align and Translate', in: *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015, Online: https://arxiv.org/abs/1409.0473.

Marco Baroni and Silvia Bernardini: 'BootCaT: Bootstrapping Corpora and Terms from the Web', in: *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pp. 1313–1316, Lisbon, Portugal, 2004, Online: http://lrec-conf.org/proceedings/lrec2004/summaries/509.htm.

Regina Barzilay and Kathleen R. McKeown: 'Sentence Fusion for Multidocument News Summarization', *Computational Linguistics* 31 (3): 297–328, 2005.

Walter Bender and Pascal Chesnais: 'Network Plus', in: *Proceedings of the SPIE Electronic Image Devices and Systems Symposium*, Los Angeles, CA, 1988.

Darina Benikova, Margot Mieskes, Christian M. Meyer, and Iryna Gurevych: 'Bridging the gap between extractive and abstractive summaries: Creation and evaluation of coherent extracts from heterogeneous sources', in: *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pp. 1039–1050, Osaka, Japan, December 2016, Online: http://aclweb.org/anthology/C16-1099.

Dominik Benz, Andreas Hotho, Robert Jäschke, Beate Krause, Folke Mitzlaff, Christoph Schmitz, and Gerd Stumme: 'The Social Bookmark and Publication Management System BibSonomy', *The VLDB Journal* 19 (6): 849–875, December 2010, Online: http://www.kde.cs.uni-kassel.de/pub/pdf/benz2010social.pdf.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein: 'Jointly Learning to Extract and Compress', in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL/HLT)*, pp. 481–490, Portland, OR, USA, 2011, Online: http://aclweb.org/anthology/P11-1049.

Shlomo Berkovsky, Timothy Baldwin, and Ingrid Zukerman: 'Aspect-Based Personalized Text Summarization', in: *Adaptive Hypermedia and Adaptive Web-Based Systems. Proceedings of the 5th International Conference*, Lecture Notes in Computer Science Vol. 5149, pp. 267–270, Springer, Berlin/Heidelberg, 2008.

Steven Bird, Robert Dale, Bonnie J. Dorr, Bryan Gibson, Mark T. Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir R. Radev, and Yee Fan Tan: 'The ACL Anthology Reference Corpus: A Reference Dataset for Bibliographic Research in Computational Linguistics', in: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, pp. 1755–1759, Marrakech, Morocco, 2008, Online: http://lrec-conf.org/proceedings/lrec2008/summaries/445.html.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan: 'Latent Dirichlet Allocation', in: *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pp. 601–608, 2001.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan: 'Latent Dirichlet Allocation', *Journal of Machine Learning Research* 3: 993–1022, 2003, Online: http://jmlr.org/papers/v3/blei03a.html.

Florian Boudin, Hugo Mougard, and Benoit Favre: 'Concept-based Summarization using Integer Linear Programming: From Concept Pruning to Multiple Optimal Solutions', in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMLNP)*, pp. 1914–1918, Lisbon, Portugal, 2015, Online: http://aclweb.org/anthology/D15-1220.

Simon Bourigault, Cédric Lagnier, Sylvain Lamprier, Ludovic Denoyer, and Patrick Gallinari: 'Learning social network embeddings for predicting information diffusion', in: *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pp. 393–402, 2014, Online: https://doi.org/10.1145/2556195.2556216.

Klaus Brinker: 'Incorporating Diversity in Active Learning with Support Vector Machines', in: *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pp. 59–66, 2003, Online: http://www.aaai.org/Library/ICML/2003/icml03-011.php.

Ziqiang Cao, Chengyao Chen, Wenjie Li, Sujian Li, Furu Wei, and Ming Zhou: 'TGSum: Build Tweet Guided Multi-Document Summarization Dataset', in: *Proceedings of the Thirtieth Conference on Artificial Intelligence (AAAI)*, pp. 2906–2912, Phoenix, AZ, USA, 2016, Online: https://aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11991.

Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei: 'Improving Multi-Document Summarization via Text Classification', in: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pp. 3053–3059, 2017, Online: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14525.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou: 'Ranking with Recursive Neural Networks and Its Application to Multi-document Summarization', in: *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence (AAAI)*, pp. 2153–2159, Austin, TX, USA, 2015a, Online: https://aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9414.

Ziqiang Cao, Furu Wei, Sujian Li, Wenjie Li, Ming Zhou, and Houfeng Wang: 'Learning Summary Prior Representation for Extractive Summarization', in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pp. 829–833, 2015b, Online: http://aclweb.org/anthology/P/P15/P15-2136.pdf.

Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou: 'Summarizing email conversations with clue words', pp. 91–100, 2007.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner: 'The AMI Meeting Corpus: A Pre-announcement', in: *Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction*, MLMI'05, pp. 28–39, Springer-Verlag, Berlin, Heidelberg, 2006, Online: http://dx.doi.org/10.1007/11677482_3.

Rose Catherine, Kathryn Mazaitis, Maxine Eskenazi, and William W. Cohen: 'Explainable Entity-based Recommendations with Knowledge Graphs', in: *Proceedings of the Poster Track of the 11th ACM Conference on Recommender Systems (RecSys 2017), Como, Italy, August 28, 2017.*, 2017.

Eugene Charniak: 'Immediate-Head Parsing for Language Models', in: *Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference, July 9-11, 2001, Toulouse, France.*, pp. 116–123, 2001, Online: http://www.aclweb.org/anthology/P01-1017.

Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma: 'Neural Attentional Rating Regression with Review-level Explanations', in: *Proceedings of the 2018 World Wide Web Conference*, WWW '18, pp. 1583–1592, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2018, Online: https://doi.org/10.1145/3178876.3186070.

Li Chen and Feng Wang: 'Sentiment-enhanced explanation of product recommendations', in: *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April*

7-11, 2014, Companion Volume, pp. 239–240, 2014, Online: https://doi.org/10.1145/2567948.2577276.

Yen-Chun Chen and Mohit Bansal: 'Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 675–686, 2018, Online: https://aclanthology.info/papers/P18-1063/p18-1063.

Jianpeng Cheng and Mirella Lapata: 'Neural Summarization by Extracting Sentences and Words', in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany, 2016, Online: https://aclweb.org/anthology/P16-1046.

Sumit Chopra, Michael Auli, and Alexander M. Rush: 'Abstractive Sentence Summarization with Attentive Recurrent Neural Networks', in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL/HLT), June 12–17, 2016, San Diego CA, USA*, pp. 93–98, 2016, Online: http://aclweb.org/anthology/N/N16/N16-1012.pdf.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio: 'Empirical evaluation of gated recurrent neural networks on sequence modeling', in: *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

James Clarke and Mirella Lapata: 'Models for Sentence Compression: A Comparison across Domains, Training Requirements and Evaluation Measures', in: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL), July 17–21, 2006, Sydney, Australia*, pp. 377–384, 2006, Online: http://aclweb.org/anthology/P06-1048.

James Clarke and Mirella Lapata: 'Modelling Compression with Discourse Constraints', in: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL), June 28–30, 2007, Prague, Czech Republic*, pp. 1–11, 2007, Online: http://www.aclweb.org/anthology/D07-1001.

James Clarke and Mirella Lapata: 'Global Inference for Sentence Compression: An Integer Linear Programming Approach', *Journal of Artificial Intelligence Research* 31: 399–429, 2008, Online: https://doi.org/10.1613/jair.2433.

J. Cohen: 'A coefficient of agreement for nominal scales', *Educational and Psychological Measurement* 20: 37–46, 1960.

David A. Cohn, Les E. Atlas, and Richard E. Ladner: 'Improving Generalization with Active Learning', *Machine Learning* 15 (2): 201–221, 1994, Online: https://doi.org/10.1007/BF00993277.

Trevor Cohn and Mirella Lapata: 'Sentence Compression Beyond Word Deletion', in: *COLING 2008, 22nd International Conference on Computational Linguistics, Proceedings of the Conference, 18-22 August 2008, Manchester, UK*, pp. 137–144, 2008, Online: http://www.aclweb.org/anthology/C08-1018.

Trevor Cohn and Mirella Lapata: 'Sentence Compression as Tree Transduction', *J. Artif. Intell. Res.* 34: 637–674, 2009, Online: https://doi.org/10.1613/jair.2655.

Trevor Cohn and Mirella Lapata: 'An Abstractive Approach to Sentence Compression', *ACM Transactions on Intelligent Systems and Technology* 4 (3): 41:1–41:35, July 2013, Online: http://doi.acm.org/10.1145/2483669.2483674.

Ronan Collobert and Jason Weston: 'A unified architecture for natural language processing: deep neural networks with multitask learning', in: *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, pp. 160–167, 2008.

John M. Conroy and Dianne P. O'Leary: 'Text Summarization via Hidden Markov Models', in: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 406–407, New Orleans, LA, USA, 2001a.

John M. Conroy and Dianne P. O'Leary: 'Text Summarization via Hidden Markov Models', in W. Bruce Croft, David J. Harper, Donald H. Kraft, and Justin Zobel (Eds.): *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA*, pp. 406–407, ACM, 2001b, Online: https://doi.org/10.1145/383952.384042.

Gordon V. Cormack: 'Email Spam Filtering: A Systematic Review', *Foundations and Trends in Information Retrieval* 1 (4): 335–455, April 2008, Online: https://doi.org/10.1561/1500000006.

Simon Corston-Oliver: 'Text Compaction for Display on Very Small Screens', in: *Proceedings of the NAACL Workshop on Automatic Summarization, June 3, 2001, Pitsburgh, PA, USA*, pp. 89–98, 2001, Online: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/naacl2001.textcompaction.corstonoliver.pdf.

William Coster and David Kauchak: 'Simple English Wikipedia: A New Text Simplification Task', in: *The 49th Annual Meeting of the Association for Computational Linguistics: Human*

Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers, pp. 665–669, 2011, Online: http://www.aclweb.org/anthology/P11-2117.

T. C. Craven: 'Abstracts produced using computer assistance.', *Journal of the American Society for Information Science* 51 (8): 745–756, 2000.

Aron Culotta and Andrew McCallum: 'Reducing Labeling Effort for Structured Prediction Tasks', in: *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pp. 746–751, 2005, Online: http://www.aaai.org/Library/AAAI/2005/aaai05-117.php.

Hoa Trang Dang: 'Overview of DUC 2005', Vancouver, Canada, October 2005.

Hoa Trang Dang: 'Overview of DUC 2006', Brooklyn, USA, June 2006.

Hoa Trang Dang and Karolina Owczarzak: 'Overview of the TAC 2008 Update Summarization Task', in: *Proceedings of the First Text Analysis Conference, TAC 2008, Gaithersburg, Maryland, USA, November 17-19, 2008*, 2008, Online: http://www.nist.gov/tac/publications/2008/additional.papers/update_summ_overview08.proceedings.pdf.

Hoa Trang Dang and Karolina Owczarzak: 'Overview of the TAC 2009 Update Summarization Task', in: *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*, 2009.

Hoa Trang Dang and Karolina Owczarzak: 'Overview of the TAC 2010 Update Summarization Task', in: *Proceedings of the Second Text Analysis Conference, TAC 2010, Gaithersburg, Maryland, USA, November 15-16, 2010*, 2010.

Hoa Trang Dang and Karolina Owczarzak: 'Overview of the TAC 2011 Update Summarization Task', in: *Proceedings of the Second Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*, 2011.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman: 'Indexing by Latent Semantic Analysis', *Journal of the American Society for Information Science* 41 (6): 391–407, 1990.

Gerald DeJong: 'Automatic Schema Acquisition in a Natural Language Environment', in: *Proceedings of the National Conference on Artificial Intelligence, Pittsburgh, PA, USA, August 18-20, 1982.*, pp. 410–413, 1982, Online: http://www.aaai.org/Library/AAAI/1982/aaai82-098.php.

Mukund Deshpande and George Karypis: 'Item-based top-N recommendation algorithms', *ACM Trans. Inf. Syst.* 22 (1): 143–177, 2004.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova: 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding', *arXiv preprint arXiv:1810.04805* 2018.

Alberto Díaz and Pablo Gervás: 'User-model Based Personalized Summarization', *Information Process Management* 43 (6): 1715–1734, nov 2007, Online: http://dx.doi.org/10.1016/j.ipm.2007.01.009.

Ted Dunning: 'Accurate Methods for the Statistics of Surprise and Coincidence', *Computational linguistics* 19 (1): 61–74, 1993, Online: https://aclweb.org/anthology/J93-1003.

Ehsan Emamjomeh-Zadeh and David Kempe: 'A General Framework for Robust Interactive Learning', in: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 7085–7094, 2017, Online: http://papers.nips.cc/paper/7283-a-general-framework-for-robust-interactive-learning.

B Endres-Niggemeyer: *Summarizing Information*, Berlin: Springer, 1998.

Martin J. Eppler and Jeanne Mengis: 'The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines', *The Information Society* 20 (5): 325–344, 2004.

Güneş Erkan and Dragomir R. Radev: 'LexRank: Graph-based Lexical Centrality As Salience in Text Summarization', *Journal of Artificial Intelligence Research* 22: 457–479, 2004, Online: https://www.aaai.org/Papers/JAIR/Vol22/JAIR-2214.pdf.

Sandra Garcia Esparza, Michael P. O'Mahony, and Barry Smyth: 'On the real-time web as a source of recommendation knowledge', in: *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, pp. 305–308, 2010.

Sandra Garcia Esparza, Michael P. O'Mahony, and Barry Smyth: 'A multi-criteria evaluation of a user generated content based recommender system', Science Foundation Ireland, 2011.

Tobias Falke: *Automatic Structured Text Summarization with Concept Maps*, Ph.D. thesis, Technische Universität, Darmstadt, 2019, Online: http://tubiblio.ulb.tu-darmstadt.de/112564/.

Thibault Févry and Jason Phang: 'Unsupervised Sentence Compression using Denoising Auto-Encoders', in: *Proceedings of the 22nd Conference on Computational Natural Language Learning, CoNLL 2018, Brussels, Belgium, October 31 - November 1, 2018*, pp. 413–422, 2018, Online: https://aclanthology.info/papers/K18-1040/k18-1040.

Katja Filippova: 'Multi-sentence Compression: Finding Shortest Paths in Word Graphs', in: *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pp. 322–330, Beijing, China, 2010, Online: https://aclweb.org/anthology/C10-1037.

Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals: 'Sentence Compression by Deletion with LSTMs', in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP) September 17–21, 2015, Lisbon, Portugal*, pp. 360–368, 2015, Online: http://aclweb.org/anthology/D/D15/D15-1042.pdf.

Katja Filippova and Yasemin Altun: 'Overcoming the Lack of Parallel Data in Sentence Compression', in: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP) October 18–21, 2013, Seattle, WA, USA*, pp. 1481–1491, 2013, Online: http://aclweb.org/anthology/D/D13/D13-1155.pdf.

Michel Galley and Kathleen R. McKeown: 'Lexicalized Markov Grammars for Sentence Compression', in: *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, pp. 180–187, 2007, Online: http://www.aclweb.org/anthology/N07-1023.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han: 'Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions', in: *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, pp. 340–348, 2010, Online: http://aclweb.org/anthology/C10-1039.

John Gantz and David Reinsel: 'The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East', *IDC Analyze the Future* 2013.

Yang Gao, Christian M. Meyer, and Iryna Gurevych: 'APRIL: Interactively Learning to Summarise by Combining Active Preference Learning and Reinforcement Learning', in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 4120–4130, 2018, Online: https://aclanthology.info/papers/D18-1445/d18-1445.

George Giannakopoulos: 'Multi-document multilingual summarization and evaluation tracks in ACL 2013 MultiLing Workshop', in: *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pp. 20–28, Association for Computational Linguistics, 2013, Online: http://aclweb.org/anthology/W13-3103.

George Giannakopoulos, Mahmoud El-Haj, Benoît Favre, Marina Litvak, Josef Steinberger, and Vasudeva Varma: 'TAC2011 MultiLing Pilot Overview', in: *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*,

2011, Online: http://www.nist.gov/tac/publications/2011/additional.papers/Summarization2011_MultiLing_overview.proceedings.pdf.

George Giannakopoulos, Jeff Kubina, John Conroy, Josef Steinberger, Benoit Favre, Mijail Kabadjov, Udo Kruschwitz, and Massimo Poesio: 'MultiLing 2015: Multilingual Summarization of Single and Multi-Documents, On-line Fora, and Call-center Conversations', in: *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 270–274, Prague, Czech Republic, 2015, Online: https://aclweb.org/anthology/W15-4638.

Dan Gillick and Benoit Favre: 'A Scalable Global Model for Summarization', in: *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pp. 10–18, Boulder, CO, USA, 2009, Online: https://aclweb.org/anthology/W09-1802.

Randy Goebel, Ajay Chander, Katharina Holzinger, Freddy Lécué, Zeynep Akata, Simone Stumpf, Peter Kieseberg, and Andreas Holzinger: 'Explainable AI: The New 42?', in: *Machine Learning and Knowledge Extraction - Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE 2018, Hamburg, Germany, August 27-30, 2018*, pp. 295–303, 2018, Online: https://doi.org/10.1007/978-3-319-99740-7_21.

David Goldberg, David A. Nichols, Brian M. Oki, and Douglas B. Terry: 'Using Collaborative Filtering to Weave an Information Tapestry', *Commun. ACM* 35 (12): 61–70, 1992, Online: https://doi.org/10.1145/138859.138867.

Kenneth Y. Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins: 'Eigentaste: A Constant Time Collaborative Filtering Algorithm', *Inf. Retr.* 4 (2): 133–151, 2001, Online: https://doi.org/10.1023/A:1011419012209.

Jade Goldstein, Vibhu O. Mittal, Jaime G. Carbonell, and James P. Callan: 'Creating and Evaluating Multi-Document Sentence Extract Summaries', in: *Proceedings of the 2000 ACM CIKM International Conference on Information and Knowledge Management, McLean, VA, USA, November 6-11, 2000*, pp. 165–172, 2000, Online: https://doi.org/10.1145/354756.354815.

Yihong Gong and Xin Liu: 'Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis', in: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 19–25, New Orleans, LA, USA, 2001.

Jesús González-Rubio, Daniel Ortiz-Martínez, and Francisco Casacuberta: 'Active Learning for Interactive Machine Translation', in: *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 245–254, Avignon, France, 2012, Online: http://aclweb.org/anthology/E12-1025.

Bryce Goodman and Seth R. Flaxman: 'European Union Regulations on Algorithmic Decision-Making and a "Right to Explanation"', *AI Magazine* 38 (3): 50–57, 2017.

Gregory Grefenstette: 'Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind', in: *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98), July 26–30, Madison, Wisconsin, USA*, pp. 111–117, 1998.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li: 'Incorporating Copying Mechanism in Sequence-to-Sequence Learning', in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, Berlin, Germany*, 2016.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio: 'Pointing the Unknown Words', in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 140–149, Association for Computational Linguistics, Berlin, Germany, August 2016, Online: http://www.aclweb.org/anthology/P16-1014.

Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su: 'The Adressa dataset for news recommendation', in: *Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, August 23-26, 2017*, pp. 1042–1048, 2017, Online: https://doi.org/10.1145/3106426.3109436.

Guibing Guo, Jie Zhang, and Neil Yorke-Smith: 'A Novel Bayesian Similarity Measure for Recommender Systems', in: *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pp. 2619–2625, 2013, Online: http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6615.

Yuhong Guo and Russell Greiner: 'Optimistic Active-Learning Using Mutual Information', in: *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pp. 823–829, 2007, Online: http://ijcai.org/Proceedings/07/Papers/132.pdf.

Iryna Gurevych, Christian M. Meyer, Carsten Binnig, Johannes Fürnkranz, Kristian Kersting, Stefan Roth, and Edwin Simpson: 'Interactive Data Analytics for the Humanities', in: *Computational Linguistics and Intelligent Text Processing - 18th International Conference, CICLing 2017, Budapest, Hungary, April 17-23, 2017, Revised Selected Papers, Part I*, pp. 527–549, 2017, Online: https://doi.org/10.1007/978-3-319-77113-7_41.

Ivan Habernal, Maria Sukhareva, Fiana Raiber, Anna Shtok, Oren Kurland, Hadar Ronen, Judit Bar-Ilan, and Iryna Gurevych: 'New Collection Announcement: Focused Retrieval Over the Web', in: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pp. 701–704, 2016, Online: http://dl.acm.org/citation.cfm?doid=2911451.2914682.

Gholamreza Haffari and Anoop Sarkar: 'Active Learning for Multilingual Statistical Machine Translation', in: *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (ACL/IJCNLP), August 2–7, 2009, Singapore*, pp. 181–189, 2009, Online: http://www.aclweb.org/anthology/P09-1021.

Aria Haghighi and Lucy Vanderwende: 'Exploring Content Models for Multi-document Summarization', in: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 362–370, Boulder, CO, USA, 2009, Online: http://aclweb.org/anthology/N09-1041.

Udo Hahn, Elena Beisswanger, Ekaterina Buyko, and Erik Faessler: 'Active Learning-Based Corpus Annotation - The PathoJen Experience', in: *American Medical Informatics Association Annual Symposium (AMIA), November 3–7, 2012, Chicago, IL, USA*, 2012, Online: http://knowledge.amia.org/amia-55142-a2012a-1.636547/t-003-1.640625/f-001-1.640626/a-038-1.641123/a-039-1.641120.

Dilek Hakkani-Tur and Gokhan Tur: 'Statistical Sentence Extraction for Information Distillation', in: *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. IV, pp. 1–4, Honolulu, HI, USA, 2007.

Andreas Hanselowski, Avinesh P. V. S., Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych: 'A Retrospective Analysis of the Fake News Challenge Stance-Detection Task', in: *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pp. 1859–1874, 2018, Online: https://aclanthology.info/papers/C18-1158/c18-1158.

Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen: 'TriRank: Review-aware Explainable Recommendation by Modeling Aspects', in: *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pp. 1661–1670, 2015.

Xiangnan He and Tat-Seng Chua: 'Neural Factorization Machines for Sparse Predictive Analytics', in: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pp. 355–364, 2017, Online: https://doi.org/10.1145/3077136.3080777.

Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua: 'Outer Product-based Neural Collaborative Filtering', in: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018,*

Stockholm, Sweden., pp. 2227–2233, 2018,
Online: https://doi.org/10.24963/ijcai.2018/308.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua: 'Neural Collaborative Filtering', in: *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pp. 173–182, 2017.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom: 'Teaching Machines to Read and Comprehend', in: *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, pp. 1693–1701, Montreal, QC, Canada, 2015, Online: https://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.pdf.

Avery Holton and Hsiang Chyi: 'News and the Overloaded Consumer: Factors Influencing Information Overload Among News Consumers', *Cyberpsychology, behavior and social networking* 15, 09 2012.

Kai Hong, John Conroy, Benoît Favre, Alex Kulesza, Hui Lin, and Ani Nenkova: 'A Repository of State of the Art and Competitive Baseline Summaries for Generic News Summarization', in: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pp. 1608–1616, Reykjavik, Iceland, 2014,
Online: http://lrec-conf.org/proceedings/lrec2014/summaries/1093.html.

Kai Hong and Ani Nenkova: 'Improving the Estimation of Word Importance for News Multi-Document Summarization', in: *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 712–721, Gothenburg, Sweden, April 2014, Online: https://aclweb.org/anthology/E14-1075.

Kazi Hoq: 'Information Overload: Causes, Consequences and Remedies - A Study', *Philosophy and Progress* 55: 49, 02 2016.

Eduard Hovy and Chin-Yew Lin: 'Automated Text Summarization and the SUMMARIST System', in Inderjeet Mani and Mark T. Maybury (Eds.): *Advances in Automatic Text Summarization*, pp. 82–94, MIT Press, 1999.

Jeremy Howard and Sebastian Ruder: 'Universal Language Model Fine-tuning for Text Classification', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 328–339, 2018, Online: https://aclanthology.info/papers/P18-1031/p18-1031.

Minqing Hu and Bing Liu: 'Mining and Summarizing Customer Reviews', in: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pp. 168–177, ACM, New York, NY, USA, 2004.

Po Hu, Donghong Ji, Chong Teng, and Yujing Guo: 'Context-Enhanced Personalized Social Summarization', in: *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pp. 1223–1238, Mumbai, India, 2012, Online: http://www.aclweb.org/anthology/C12-1075.

Yifan Hu, Yehuda Koren, and Chris Volinsky: 'Collaborative Filtering for Implicit Feedback Datasets', in: *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pp. 263–272, 2008, Online: https://doi.org/10.1109/ICDM.2008.22.

Aishwarya Jadhav and Vaibhav Rajan: 'Extractive Summarization with SWAP-NET: Sentences and Words from Alternating Pointer Networks', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pp. 142–151, 2018, Online: https://aclanthology.info/papers/P18-1014/p18-1014.

Hongyan Jing: 'Sentence Reduction for Automatic Text Summarization', in: *6th Applied Natural Language Processing Conference, ANLP 2000, Seattle, Washington, USA, April 29 - May 4, 2000*, pp. 310–315, 2000, Online: http://aclweb.org/anthology/A/A00/A00-1043.pdf.

Hidetaka Kamigaito, Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata: 'Higher-Order Syntactic Attention Network for Longer Sentence Compression', in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL/HLT), June 1–6, 2018, New Orleans, LA, USA*, pp. 1716–1726, 2018, Online: https://aclanthology.info/papers/N18-1155/n18-1155.

Martin Kay: 'The Proper Place of Men and Machines in Language Translation.', *Machine Translation* 12 (1-2): 3–23, 1997.

Chris Kedzie, Kathleen R. McKeown, and Hal Daumé III: 'Content Selection in Deep Learning Models of Summarization', in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1818–1828, Brussels, Belgium, 2018, Online: https://aclweb.org/anthology/D18-108.

Daniel A. Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon: 'Visual Analytics: Definition, Process, and Challenges', in Andreas Kerren et al. (Eds.): *Information Visualization*, Lecture Notes in Computer Science Vol. 4950, pp. 154–175, Springer, 2008.

Ryan Kelly: 'Twitter study (White paper)', 2009, Online: http://www.pearanalytics.com/wp-content/uploads/2012/12/Twitter-Study-August-2009.pdf.

Atif Khan, Naomie Salim, and Haleem Farman: 'Clustered genetic semantic graph approach for multi-document abstractive summarization', in: *International Conference on Intelligent Systems Engineering (ICISE)*, pp. 63–70, Islamabad, Pakistan, 01 2016.

Benjamin Kille, Frank Hopfgartner, Torben Brodt, and Tobias Heintz: 'The Plista Dataset', in: *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*, NRS '13, pp. 16–23, ACM, New York, NY, USA, 2013, Online: http://doi.acm.org/10.1145/2516641.2516643.

Yoon Kim: 'Convolutional Neural Networks for Sentence Classification', in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, 2014, Online: https://aclweb.org/anthology/D14-1181.

Diederik P. Kingma and Jimmy Lei Ba: 'Adam: A Method for Stochastic Optimization', in: *Proceedings of the 3rd International Conference on Learning Representations (ICLR), May 7–9, 2015, San Diego, CA, USA*, 2015, Online: https://arxiv.org/abs/1412.6980.

Dan Klein and Christopher D. Manning: 'Accurate Unlexicalized Parsing', in: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL)*, pp. 423–430, Sapporo, Japan, 2003, Online: http://www.aclweb.org/anthology/P03-1054.

Kevin Knight and Daniel Marcu: 'Statistics-based Summarization — Step One: Sentence Compression', in: *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI)*, pp. 703–710, Austin, TX, USA, 2000, Online: https://aaai.org/Papers/AAAI/2000/AAAI00-108.pdf.

Kevin Knight and Daniel Marcu: 'Summarization beyond sentence extraction: A probabilistic approach to sentence compression', *Artificial Intelligence* 139 (1): 91–107, 2002, Online: https://doi.org/10.1016/S0004-3702(02)00222-9.

Rebecca Knowles and Philipp Koehn: 'Neural Interactive Translation Prediction', in: *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*, 2016.

Youngjoong Ko and Jungyun Seo: 'An Effective Sentence-Extraction Technique using Contextual Information and Statistical Approaches for Text Summarization', *Pattern Recognition Letters* 29 (9): 1366–1371, 2008.

Philipp Koehn, Franz Josef Och, and Daniel Marcu: 'Statistical Phrase-Based Translation', in: *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL), May 27–June 1, 2003, Edmonton, Canada*, pp. 48–54, 2003, Online: http://aclweb.org/anthology/N/N03/N03-1017.pdf.

Varada Kolhatkar and Maite Taboada: 'Using New York Times Picks to Identify Constructive Comments', in: *Proceedings of the Second Workshop on Natural Language Processing meets Journalism*, pp. 100–105, Copenhagen, Denmark, 2017.

Yehuda Koren: 'Factorization meets the neighborhood: a multifaceted collaborative filtering model', in: *Proceedings of the 14th ACM SIGKDD, International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pp. 426–434, 2008.

Bill Kovach and Tom Rosenstiel: *The Elements of Journalism: What Newspeople Should Know and the Public Should Expect*, New York: Three Rivers Press, 2007.

Jan Kremer, Kim Steenstrup Pedersen, and Christian Igel: 'Active learning with support vector machines', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4 (4): 313–326, 2014, Online: http://dx.doi.org/10.1002/widm.1132.

Julian Kupiec, Jan Pedersen, and Francine Chen: 'A Trainable Document Summarizer', in: *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 68–73, Seattle, WA, USA, 1995.

Daniel D. Lee and H. Sebastian Seung: 'Algorithms for Non-negative Matrix Factorization', in: *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pp. 556–562, 2000.

Cane Wing-ki Leung, Stephen Chan, and Fu-Lai Chung: 'Integrating Collaborative Filtering and Sentiment Analysis: A Rating Inference Approach', pp. 62–66, 01 2006.

David D. Lewis and William A. Gale: 'A Sequential Algorithm for Training Text Classifiers', in: *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994 (Special Issue of the SIGIR Forum)*, pp. 3–12, 1994, Online: http://dl.acm.org/citation.cfm?id=188495.

Chen Li, Xian Qian, and Yang Liu: 'Using Supervised Bigram-based ILP for Extractive Summarization', in: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1004–1013, Sofia, Bulgaria, 2013, Online: http://aclweb.org/anthology/P13-1099.

Chenliang Li, Weiran Xu, Si Li, and Sheng Gao: 'Guiding Generation for Abstractive Text Summarization Based on Key Information Guide Network', in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pp. 55–60, 2018, Online: https://aclanthology.info/papers/N18-2009/n18-2009.

Liangda Li, Ke Zhou, Gui-Rong Xue, Hongyuan Zha, and Yong Yu: 'Enhancing Diversity, Coverage and Balance for Summarization Through Structure Learning', in: *Proceedings of the 18th International Conference on World Wide Web (WWW)*, pp. 71–80, Madrid, Spain, 2009.

Piji Li, Lidong Bing, and Wai Lam: 'Reader-Aware Multi-Document Summarization: An Enhanced Model and The First Dataset', in: *Proceedings of the EMNLP Workshop on New Frontiers in Summarization*, pp. 91–99, Copenhagen, Denmark, 2017a, Online: https://aclweb.org/anthology/W17-4512.

Piji Li, Zihao Wang, Lidong Bing, and Wai Lam: 'Persona-Aware Tips Generation', *arXiv e-prints* arXiv:1903.02156, 2019.

Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam: 'Neural Rating Regression with Abstractive Tips Generation for Recommendation', in: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 345–354, Shinjuku, Tokyo, Japan, 2017b.

Yang Li, Ying Lv, Suge Wang, Jiye Liang, Juanzi Li, and Xiaoli Li: 'Cooperative Hybrid Semi-Supervised Learning for Text Sentiment Classification', *Symmetry* 11 (2): 133, 2019, Online: https://doi.org/10.3390/sym11020133.

Chin-Yew Lin: 'ROUGE: A Package for Automatic Evaluation of Summaries', in: *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pp. 74–81, Barcelona, Spain, July 2004, Online: http://aclweb.org/anthology/W04-1013.

Chin-Yew Lin and Eduard Hovy: 'Manual and Automatic Evaluation of Summaries', in: *Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4*, AS '02, pp. 45–51, Association for Computational Linguistics, 2002, Online: https://doi.org/10.3115/1118162.1118168.

Jimmy Lin, Adam Roegiest, Luchen Tan, Richard McCreadie, Ellen Voorhees, and Fernando Diaz: 'Overview of the TREC 2016 Real-Time Summarization Track', in: *Proceedings of The Twenty-Fifth Text REtrieval Conference (TREC)*, Gaithersburg, MD, USA, 2016, Online: http://trec.nist.gov/pubs/trec25/papers/Overview-RT.pdf.

Bing Liu: 'Sentiment Analysis and Subjectivity', in: *Handbook of Natural Language Processing, Second Edition.*, pp. 627–666, Boca Raton: CRC Press, 2010.

Jiahui Liu, Elin Pedersen, and Peter Dolan: 'Personalized News Recommendation Based on Click Behavior', in: *International Conference on Intelligent User Interfaces*, 2010.

Linqing Liu, Yao Lu, Min Yang, Qiang Qu, Jia Zhu, and Hongyan Li: 'Generative Adversarial Network for Abstractive Text Summarization', in: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 8109–8110, 2018a, Online: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16238.

Ming Liu, Wray L. Buntine, and Gholamreza Haffari: 'Learning to Actively Learn Neural Machine Translation', in: *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL), October 31–November 1, 2018, Brussels, Belgium*, pp. 334–344, 2018b, Online: https://aclanthology.info/papers/K18-1033/k18-1033.

Nathan N. Liu, Xiangrui Meng, Chao Liu, and Qiang Yang: 'Wisdom of the Better Few: Cold Start Recommendation via Representative Based Rating Elicitation', in: *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pp. 37–44, ACM, New York, NY, USA, 2011, Online: http://doi.acm.org/10.1145/2043932.2043943.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang: 'Adversarial Multi-task Learning for Text Classification', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1–10, 2017.

Ying Liu: 'Active Learning with Support Vector Machine Applied to Gene Expression Data for Cancer Classification', *Journal of Chemical Information and Modeling* 44 (6): 1936–1941, 2004, Online: https://doi.org/10.1021/ci049810a.

Zhicheng Liu and Jeffrey Heer: 'The effects of interactive latency on exploratory visual analysis', *IEEE transactions on visualization and computer graphics* 20: 2122–2131, 2014.

Elena Lloret and Manuel Palomar: 'Towards automatic tweet generation: A comparative study from the text summarization perspective in the journalism genre', *Expert Systems with Applications* 40 (16): 6624–6630, 2013.

Edward Loper and Steven Bird: 'NLTK: The Natural Language Toolkit', in: *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pp. 63–70, 2002, Online: http://dx.doi.org/10.3115/1118108.1118117.

Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro: *Content-based Recommender Systems: State of the Art and Trends*, pp. 73–105, Springer US, Boston, MA, 2011, Online: https://doi.org/10.1007/978-0-387-85820-3_3.

Claude de Loupy, Marie Guégan, Christelle Ayache, Somara Seng, and Juan-Manuel Torres-Moreno: 'A French Human Reference Corpus for Multi-Document Summarization and Sentence Compression', in: *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*, 2010, Online: http://www.lrec-conf.org/proceedings/lrec2010/summaries/919.html.

H. P. Luhn: 'The Automatic Creation of Literature Abstracts', *IBM Journal of Research and Development* 2 (2): 159–165, 1958, Online: http://www.research.ibm.com/journal/rd/022/luhn.pdf.

Thang Luong, Hieu Pham, and Christopher D. Manning: 'Effective Approaches to Attention-based Neural Machine Translation', in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), September 17–21, 2015, Lisbon, Portugal*, pp. 1412–1421, 2015, Online: http://aclweb.org/anthology/D/D15/D15-1166.pdf.

Juhani Luotolahti and Filip Ginter: 'Sentence Compression For Automatic Subtitling', in: *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA), May 11–13, 2015, Vilnius, Lithuania*, pp. 135–143, 2015, Online: http://aclweb.org/anthology/W/W15/W15-1818.pdf.

Duy Khang Ly, Kazunari Sugiyama, Ziheng Lin, and Min-Yen Kan: 'Product Review Summarization based on Facet Identification and Sentence Clustering', *CoRR* abs/1110.1428, 2011.

François Mairesse, Milica Gašić, Filip Jurčıček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve J. Young: 'Phrase-Based Statistical Language Generation Using Graphical Models and Active Learning', in: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), July 11–16, 2010, Uppsala, Sweden*, pp. 1552–1561, 2010, Online: http://www.aclweb.org/anthology/P10-1157.

Inderjeet Mani and Eric Bloedorn: 'Multi-document Summarization by Graph Search and Matching', in: *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pp. 622–628, Providence, RI, USA, 1997, Online: https://aaai.org/Papers/AAAI/1997/AAAI97-097.pdf.

Inderjeet Mani and Eric Bloedorn: 'Summarizing Similarities and Differences among Related Documents', *Information Retrieval* 1 (1-2): 35–67, 1999.

Leandro Balby Marinho, Alexandros Nanopoulos, Lars Schmidt-Thieme, Robert Jáschke, Andreas Hotho, Gerd Stumme, and Panagiotis Symeonidis: 'Social Tagging Recommender Systems', in: *Recommender Systems Handbook*, pp. 615–644, 2011, Online: https://doi.org/10.1007/978-0-387-85820-3_19.

Benjamin M. Marlin: 'Modeling User Rating Profiles For Collaborative Filtering', in: *Proceedings of NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pp. 627–634, 2003.

Erwin Marsi, Emiel Krahmer, Iris Hendrickx, and Walter Daelemans: 'On the Limits of Sentence Compression by Deletion', in: *Empirical Methods in Natural Language Generation: Data-oriented Methods and Empirical Evaluation*, Lecture Notes in Artificial Intelligence Vol. 5790, pp. 45–66, 2010, Online: https://doi.org/10.1007/978-3-642-15573-4_3.

Julian J. McAuley and Jure Leskovec: 'Hidden factors and hidden topics: understanding rating dimensions with review text', in: *Proceedings of the 7th ACM RecSys '13, Hong Kong, China, October 12-16*, pp. 165–172, 2013.

Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel: 'Image-Based Recommendations on Styles and Substitutes', in: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pp. 43–52, 2015, Online: https://doi.org/10.1145/2766462.2767755.

Ryan McDonald: 'A Study of Global Inference Algorithms in Multi-document Summarization', in: *Advances in Information Retrieval. Proceedings of the 29th European Conference on IR Research (ECIR)*, Lecture Notes in Computer Science Vol. 4425, pp. 557–564, Springer, Berlin/Heidelberg, 2007, Online: http://dl.acm.org/citation.cfm?id=1763653.1763720.

Ryan T. McDonald: 'Discriminative Sentence Compression with Soft Syntactic Evidence', in: *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*, 2006, Online: http://aclweb.org/anthology/E/E06/E06-1038.pdf.

Simon McEnnis: 'Following the action: How live bloggers are reimagining the professional ideology of sports journalism', *Journalism Practice* 10 (8): 967–982, 2016.

Yishu Miao and Phil Blunsom: 'Language as a Latent Variable: Discrete Generative Models for Sentence Compression', in: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), November 1–4, 2016, Austin, TX, USA*, pp. 319–328, 2016, Online: http://aclweb.org/anthology/D/D16/D16-1031.pdf.

Rada Mihalcea and Paul Tarau: 'TextRank: Bringing Order into Text', in: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 404–411, Barcelona, Spain, July 2004, Online: http://aclweb.org/anthology/W04-3252.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean: 'Efficient Estimation of Word Representations in Vector Space', *CoRR* abs/1301.3781, 2013, Online: http://arxiv.org/abs/1301.3781.

Subhabrata Mukherjee, Kashyap Popat, and Gerhard Weikum: 'Exploring Latent Semantic Factors to Find Useful Product Reviews', in: *Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, Texas, USA, April 27-29, 2017.*, pp. 480–488, 2017.

Claudiu Cristian Musat, Yizhong Liang, and Boi Faltings: 'Recommendation Using Textual Opinions', in: *Proceedings of the 23rd IJCAI, Beijing, China, August 3-9*, pp. 2684–2690, 2013.

Masahiro Nakano, Hideyuki Shibuki, Rintaro Miyazaki, Madoka Ishioroshi, Koichi Kaneko, and Tatsunori Mori: 'Construction of Text Summarization Corpus for the Credibility of Information on the Web', in: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, pp. 3125–3131, Valletta, Malta, 2010.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou: 'SummaRuNNer: A Recurrent Neural Network Based Sequence Model for Extractive Summarization of Documents', in: *Proceedings of the Thirty-First Conference on Artificial Intelligence (AAAI)*, pp. 3075–3081, San Francisco, CA, USA, 2017, Online: https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14636/14080.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang: 'Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond', in: *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pp. 280–290, Berlin, Germany, 2016, Online: https://aclweb.org/anthology/K16-1028.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme: 'Annotated Gigaword', in: *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX), June 7–8, 2012, Montréal, Canada*, pp. 95–100, 2012, Online: http://dl.acm.org/citation.cfm?id=2391200.2391218.

Shashi Narayan and Claire Gardent: 'Hybrid Simplification using Deep Semantics and Machine Translation', in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pp. 435–445, 2014, Online: http://aclweb.org/anthology/P/P14/P14-1041.pdf.

Shashi Narayan, Nikos Papasarantopoulos, Mirella Lapata, and Shay B. Cohen: 'Neural Extractive Summarization with Side Information', *CoRR* abs/1704.04530, 2017, Online: http://arxiv.org/abs/1704.04530.

Masumi Narita, Kazuya Kurokawa, and Takehito Utsuro: 'A Web-based English Abstract Writing Tool Using a Tagged E–J Parallel Corpus', in: *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, Las Palmas, Spain, 2002, Online: http://www.lrec-conf.org/proceedings/lrec2002/sumarios/137.htm.

Ani Nenkova and Annie Louis: 'Can You Summarize This? Identifying Correlates of Input Difficulty for Multi-Document Summarization', in: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 825–833, Columbus, OH, USA, 2008, Online: https://aclweb.org/anthology/P08-1094.

Ani Nenkova and Kathleen R. McKeown: 'A Survey of Text Summarization Techniques', in Charu C. Aggarwal and ChengXiang Zhai (Eds.): *Mining Text Data*, pp. 43–76, Boston: Springer, 2012.

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown: 'A Compositional Context Sensitive Multi-document Summarizer: Exploring the Factors That Influence Summarization', in: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 573–580, Seattle, WA, USA, 2006.

Laurent Nepveu, Guy Lapalme, Philippe Langlais, and George F. Foster: 'Adaptive Language and Translation Models for Interactive Machine Translation', in: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pp. 190–197, 2004, Online: http://www.aclweb.org/anthology/W04-3225.

Daan Odijk and Anne Schuth: 'Online Learning to Rank for Recommender Systems', in: *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pp. 348–348, ACM, New York, NY, USA, 2017, Online: http://doi.acm.org/10.1145/3109859.3109925.

Constantin Orăsan and Laura Hasler: 'Computer-aided Summarisation: What the User Really Wants', in: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pp. 1548–1551, Genoa, Italy, 2006, Online: http://www.lrec-conf.org/proceedings/lrec2006/summaries/52.html.

Constantin Orăsan, Ruslan Mitkov, and Laura Hasler: 'CAST: a computer-aided summarisation tool', in: *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics (EACL)*, pp. 135–138, Budapest, Hungary, 2003, Online: http://aclweb.org/anthology/E03-1066.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato: 'Analyzing Uncertainty in Neural Machine Translation', in: *Proceedings of the 35th International Conference on*

Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018, pp. 3953–3962, 2018, Online: http://proceedings.mlr.press/v80/ott18a.html.

Paul Over, Hoa Dang, and Donna Harman: 'DUC in Context', *Inf. Process. Manage.* 43 (6): 1506–1520, November 2007, Online: http://dx.doi.org/10.1016/j.ipm.2007.01.019.

Paul Over and Walter Liggett: 'Introduction to DUC-2002: an Intrinsic Evaluation of Generic News Text Summarization Systems', Philadelphia, USA, July 2002.

Paul Over and James Yen: 'Introduction to DUC-2001: an Intrinsic Evaluaton of Generic News Text Summarization Systems', New Orleans, USA, September 2001.

Paul Over and James Yen: 'An Introduction to DUC 2003 Intrinsic Evaluation of Generic News Text Summarization Systems', Edmonton, Canada, May 2003.

Paul Over and James Yen: 'An Introduction to DUC 2004 Intrinsic Evaluation of Generic New Text Summarization Systems', Boston, USA, May 2004.

Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova: 'An Assessment of the Accuracy of Automatic Evaluation in Summarization', in: *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pp. 1–9, Montréal, Canada, 2012, Online: http://aclweb.org/anthology/W12-2601.

Sun Park and Dong Un An: 'Automatic Query-based Personalized Summarization That Uses Pseudo Relevance Feedback with NMF', in: *Proceedings of the 4th International Conference on Ubiquitous Information Management and Communication (ICUIMC)*, pp. 61:1–61:7, 2010, Online: http://doi.acm.org/10.1145/2108616.2108690.

E. S. Patterson, E. M. Roth, and D. D. Woods: 'Predicting Vulnerabilities in Computer-Supported Inferential Analysis under Data Overload', *Cognition, Technology & Work* 3 (4): 224–237, 2001.

John Pavlopoulos, Prodromos Malakasiotis, Juli Bakagianni, and Ion Androutsopoulos: 'Improved Abusive Comment Moderation with User Embeddings', in: *Proceedings of the Second Workshop on Natural Language Processing meets Journalism*, pp. 51–55, Copenhagen, Denmark, 2017.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning: 'Glove: Global Vectors for Word Representation', in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, 2014, Online: https://aclweb.org/anthology/D14-1162.

Juan Antonio Pérez-Ortiz, Daniel Torregrosa, and Mikel L. Forcada: 'Black-box integration of heterogeneous bilingual resources into an interactive translation system', in: *Proceedings of the Workshop on Humans and Computer-assisted Translation, HaCaT@EACL 2014, Gothenburg, Sweden, April 26, 2014*, pp. 57–65, 2014, Online: https://doi.org/10.3115/v1/W14-0309.

Álvaro Peris and Francisco Casacuberta: 'Active Learning for Interactive Neural Machine Translation of Data Streams', in: *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL), October 31–November 1, 2018, Brussels, Belgium*, pp. 151–160, 2018, Online: https://aclanthology.info/papers/K18-1015/k18-1015.

Helen Petrie and Nigel Bevan: 'The Evaluation of Accessibility, Usability, and User Experience', in Constantine Stephanidis (Ed.): *The Universal Access Handbook*, Human Factors and Ergonomics, chapter 20, pp. 1–16, Boca Raton: CRC Press, 2009, Online: https://www.crcpress.com/product/isbn/9780805862805.

Maxime Peyrard and Judith Eckle-Kohler: 'Optimizing an Approximation of ROUGE – a Problem-Reduction Approach to Extractive Multi-Document Summarization', in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1825–1836, Berlin, Germany, 2016, Online: https://aclweb.org/anthology/P16-1172.

Maxime Peyrard and Judith Eckle-Kohler: 'Supervised Learning of Automatic Pyramid for Optimization-Based Multi-Document Summarization', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, Vol. 1, pp. 1084–1094, Vancouver, BC, Canada, 2017.

John C. Platt: 'Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods', in: *Advances In Large Margin Classifiers*, pp. 61–74, MIT Press, 1999.

Odette Pollar: *Surviving Information Overload How to Find, Filter, and Focus on What's Important*, Crisp Learning, 2004.

Octavian Popescu and Carlo Strapparava (Eds.): *Proceedings of the Second Workshop on Natural Language Processing meets Journalism*, Copenhagen, Denmark, 2017, Online: https://aclweb.org/anthology/W17-4200.

Octavian Popescu and Carlo Strapparava (Eds.): *Proceedings of the Third Workshop on Natural Language Processing meets Journalism*, Miyazaki, Japan, 2018, Online: http://lrec-conf.org/workshops/lrec2018/W13/.

Mickaël Poussevin, Vincent Guigue, and Patrick Gallinari: 'Extended Recommendation Framework: Generating the Text of a User Review as a Personalized Summary', in: *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015*, pp. 34–41, 2015, Online: http://ceur-ws.org/Vol-1448/paper7.pdf.

Avinesh P.V.S., Carsten Binnig, Benjamin Hättasch, Christian Meyer, and Orkan Özyurt: 'Sherlock: A System for Interactive Summarization of Large Text Collections', *PVLDB* 11 (12): 1902–1905, 2018a, Online: http://www.vldb.org/pvldb/vol11/p1902-p.v.s..pdf.

Avinesh P.V.S. and Christian M. Meyer: 'Joint Optimization of User-desired Content in Multi-document Summaries by Learning from User Feedback', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, Vol. Volume 1: Long Paper, pp. 1353–1363, Association for Computational Linguistics, July 2017.

Avinesh P.V.S. and Christian M. Meyer: 'Data-efficient Neural Text Compression with Interactive Learning', in: *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, June 2019. (to appear).

Avinesh P.V.S., Maxime Peyrard, and Christian M. Meyer: 'Live Blog Corpus for Summarization', in: *Proceedings of the 11th Language Resources and Evaluation Conference*, European Language Resource Association, Miyazaki, Japan, May 2018b, Online: https://www.aclweb.org/anthology/L18-1505.

Avinesh P.V.S., Yongli Ren, Christian M. Meyer, Jeffrey Chan, Zhifeng Bao, and Mark Sanderson: 'J3R: Joint Multi-task Learning of Ratings and Review Summaries for Explainable Recommendation', in: *Proceedings of The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2019), Würzburg, Germany, September 16-20, 2019*, Springer, August 2019, Online: http://tubiblio.ulb.tu-darmstadt.de/115764/.

Dragomir Radev, Jahna Otterbacher, Adam Winkel, and Sasha Blair-Goldensohn: 'NewsInEssence: Summarizing Online News Topics', *Commun. ACM* 48 (10): 95–98, October 2005, Online: https://doi.org/10.1145/1089107.1089111.

Dragomir R. Radev, Eduard Hovy, and Kathleen R. McKeown: 'Introduction to the Special Issue on Summarization', *Computational Linguistics* 28 (4): 399–408, dec 2002.

Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska: 'Centroid-based Summarization of Multiple Documents: Sentence Extraction, Utility-based Evaluation, and

User Studies', in: *Proceedings of the NAACL-ANLP Workshop on Automatic Summarization*, pp. 21–30, Seattle, Washington, 2000, Online: https://aclweb.org/anthology/W00-0403.

Sindhu Raghavan, Suriya Gunasekar, and Joydeep Ghosh: 'Review quality aware collaborative filtering', in: *Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012*, pp. 123–130, 2012, Online: https://doi.org/10.1145/2365952.2365978.

G. J. Rath, A. Resnick, and T. R. Savage: 'The formation of abstracts by the selection of sentences. Part I. Sentence selection by men and machines', *American Documentation* 12 (2): 139–141, 1961, Online: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.5090120210.

Marek Rei: 'Semi-supervised Multitask Learning for Sequence Labeling', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL, 2017, Vancouver, Canada, July 30 - August 4*, pp. 2121–2130, 2017.

Nils Reimers, Judith Eckle-Kohler, Carsten Schnober, Jungi Kim, and Iryna Gurevych: 'GermEval-2014: Nested Named Entity Recognition with Neural Networks', in: *Workshop Proceedings of the 12th Edition of the KONVENS Conference*, pp. 117–120, Hildesheim, Germany, October 2014.

Pengjie Ren, Zhumin Chen, Zhaochun Ren, Furu Wei, Liqiang Nie, Jun Ma, and Maarten de Rijke: 'Sentence Relations for Extractive Summarization with Deep Neural Networks', *ACM Trans. Inf. Syst.* 36 (4): 39:1–39:32, 2018, Online: https://doi.org/10.1145/3200864.

Pengjie Ren, Furu Wei, Zhumin Chen, Jun Ma, and Ming Zhou: 'A Redundancy-Aware Sentence Regression Framework for Extractive Summarization', in: *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pp. 33–43, 2016, Online: http://aclweb.org/anthology/C/C16/C16-1004.pdf.

Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke: 'Social Collaborative Viewpoint Regression with Explainable Recommendations', in: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017*, pp. 485–494, 2017.

Dan Roth: 'Incidental Supervision: Moving beyond Supervised Learning', in: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pp. 4885–4890, 2017, Online: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14950.

Jan Rupnik, Andrej Muhič, Gregor Leban, Primož Škraba, Blaž Fortuna, and Marko Grobelnik: 'News across Languages - Cross-Lingual Document Similarity and Event Tracking', *J. Artif. Int. Res.* 55 (1): 283–316, January 2016.

Alexander M. Rush, Sumit Chopra, and Jason Weston: 'A Neural Attention Model for Abstractive Sentence Summarization', in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), September 17–21, 2015, Lisbon, Portugal*, pp. 379–389, 2015, Online: http://aclweb.org/anthology/D/D15/D15-1044.pdf.

Horacio Saggion and Guy Lapalme: 'Generating Indicative-informative Summaries with sumUM', *Comput. Linguist.* 28 (4): 497–526, December 2002, Online: http://dx.doi.org/10.1162/089120102762671963.

Ruslan Salakhutdinov and Andriy Mnih: 'Probabilistic Matrix Factorization', in: *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pp. 1257–1264, 2007.

Abigail See, Peter J. Liu, and Christopher D. Manning: 'Get To The Point: Summarization with Pointer-Generator Networks', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1073–1083, Vancouver, BC, Canada, 2017.

Ozan Sener and Silvio Savarese: 'Active Learning for Convolutional Neural Networks: A Core-Set Approach', in: *Proceedings of the 6th International Conference on Learning Representations (ICLR), May 6–9, New Orleans, LA, USA*, 2018, Online: https://arxiv.org/pdf/1708.00489.pdf.

Yanir Seroussi, Fabian Bohnert, and Ingrid Zukerman: 'Personalised rating prediction for new users using latent factor models', in: *HT'11, Proceedings of the 22nd ACM Conference on Hypertext and Hypermedia, Eindhoven, The Netherlands, June 6-9, 2011*, pp. 47–56, 2011, Online: https://doi.org/10.1145/1995966.1995976.

Burr Settles: *Active Learning*, Morgan & Claypool Publishers, 2012.

Burr Settles and Mark Craven: 'An Analysis of Active Learning Strategies for Sequence Labeling Tasks', in: *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1070–1079, 2008, Online: http://www.aclweb.org/anthology/D08-1112.

C. E. Shannon: ‘A Mathematical Theory of Communication’, *Bell System Technical Journal* 27 (3): 379–423, 1948,
Online: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01338.x.

Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew Lim Tan: ‘Multi-Criteria-based Active Learning for Named Entity Recognition’, in: *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain.*, pp. 589–596, 2004, Online: http://aclweb.org/anthology/P/P04/P04-1075.pdf.

Stuart M. Shieber and Yves Schabes: ‘Synchronous Tree-Adjoining Grammars’, in: *13th International Conference on Computational Linguistics, COLING 1990, University of Helsinki, Finland, August 20-25, 1990*, pp. 253–258, 1990,
Online: http://aclweb.org/anthology/C90-3045.

Edwin D. Simpson and Iryna Gurevych: ‘Finding Convincing Arguments Using Scalable Bayesian Preference Learning’, *TACL* 6: 357–371, 2018,
Online: https://transacl.org/ojs/index.php/tacl/article/view/1304.

Anders Søgaard and Yoav Goldberg: ‘Deep multi-task learning with low level tasks supervised at lower layers’, in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), August 7–12, 2016, Berlin, Germany*, 2016,
Online: http://aclweb.org/anthology/P/P16/P16-2038.pdf.

Karen Sparck Jones: ‘A Statistical Interpretation of Term Specificity and its Application in Retrieval’, *Journal of Documentation* 28 (1): 11–21, 1972.

Josef Steinberger and Karel Jezek: ‘Using latent semantic analysis in text summarization and summary evaluation’, in: *Proceedings of the 7th International Conference on Information Systems Implementation and Modelling (ISIM)*, pp. 93–100, Rožnov pod Radhoštěm, Czech Republic, 2004.

Ralf Steinberger: ‘Multilingual and Cross-Lingual News Analysis in the Europe Media Monitor (EMM) (Extended Abstract)’, in Mihai Lupu, Evangelos Kanoulas, and Fernando Loizides (Eds.): *Multidisciplinary Information Retrieval*, pp. 1–4, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

Bipin Suresh: ‘Inclusion of large input corpora in Statistical Machine Translation’, *Technical report*, Stanford University, 2010,
Online: https://nlp.stanford.edu/courses/cs224n/2010/reports/bipins.pdf.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le: ‘Sequence to Sequence Learning with Neural Networks’, in: *Proceedings of the 27th International Conference on Neural Information*

Processing Systems (NIPS), pp. 3104–3112, Montreal, QC, Canada, 2014, Online: https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Hiroya Takamura and Manabu Okumura: 'Learning to generate summary as structured output', in: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 1437–1440, Toronto, QC, Canada, 2010.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata: 'Neural Headline Generation on Abstract Meaning Representation', in: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1054–1059, Association for Computational Linguistics, Austin, Texas, November 2016, Online: https://www.aclweb.org/anthology/D16-1112.

Christopher Tauchmann, Thomas Arnold, Andreas Hanselowski, Christian M. Meyer, and Margot Mieskes: 'Beyond Generic Summarization: A Multi-faceted Hierarchical Summarization Corpus of Large Heterogeneous Data', in: *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*, pp. 3184–3191, Miyazaki, Japan, May 2018, Online: http://lrec-conf.org/proceedings/lrec2018/summaries/252.html.

Einar Thorsen: 'Live Blogging and Social Media Curation: Challenges and Opportunities for Journalism', in Karen Fowler-Watt and Stuart Allan (Eds.): *Journalism: New Challenges*, chapter 8, pp. 123–145, Poole: Centre for Journalism & Communication Research, Bournemouth University, 2013, Online: http://eprints.bournemouth.ac.uk/20926/.

Einar Thorsen and Daniel Jackson: 'Seven Characteristics Defining Online News Formats: Towards a typology of online news and live blogs', *Digital Journalism* 6 (7): 847–868, 2018.

Neil Thurman and Nic Newman: 'The Future of Breaking News Online? A study of live blogs through surveys of their consumption, and of readers' attitudes and participation', *Journalism Studies* 15 (5): 655–667, 2014.

Neil Thurman and Aljosha Karim Schapals: 'Live blogs, sources, and objectivity: The contradictions of real-time online reporting', in: *The Routledge Companion to Digital Journalism Studies*, pp. 283–292, London/New York: Routledge, 2017.

Neil Thurman and Anna Walters: 'Live Blogging — Digital Journalism's Pivotal Platform? A case study of the production, consumption, and form of Live Blogs at Guardian.co.uk', *Digital Journalism* 1 (1): 82–101, 2013.

Nava Tintarev and Judith Masthoff: 'Designing and Evaluating Explanations for Recommender Systems', in: *Recommender Systems Handbook*, pp. 479–510, 2011.

Juan-Manuel Torres-Moreno: *Automatic Text Summarization*, John Wiley Sons, Inc, 2014.

Kristina Toutanova, Chris Brockett, Ke M. Tran, and Saleema Amershi: 'A Dataset and Evaluation Metrics for Abstractive Compression of Sentences and Short Paragraphs', in: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), November 1-4, 2016, Austin, TX, USA*, pp. 340–350, 2016, Online: http://aclweb.org/anthology/D/D16/D16-1033.pdf.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun: 'Large Margin Methods for Structured and Interdependent Output Variables', *Journal of Machine Learning Research* 6: 1453–1484, 2005, Online: http://jmlr.org/papers/v6/tsochantaridis05a.html.

Jenine Turner and Eugene Charniak: 'Supervised and Unsupervised Learning for Sentence Compression', in: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), June 25-30, 2005, Ann Arbor, MI, USA*, pp. 290–297, 2005, Online: http://aclweb.org/anthology/P/P05/P05-1036.pdf.

J. Ulrich, G. Murray, and G. Carenini: 'A Publicly Available Annotated Corpus for Supervised Email Summarization', in: *AAAI08 EMAIL Workshop*, AAAI, Chicago, USA, 2008.

Vincent Vandegehinste and Yi Pan: 'Sentence compression for automated subtitling: A hybrid approach', in: *Proceedings of the ACL Workshop "Text Summarization Branches Out", July 25-26, 2004, Barcelona, Spain*, pp. 89–95, 2004, Online: http://www.aclweb.org/anthology/W04-1015.

Paolo Viappiani and Craig Boutilier: 'Optimal Bayesian Recommendation Sets and Myopically Optimal Choice Query Sets', in: *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pp. 2352–2360, 2010, Online: http://papers.nips.cc/paper/3943-optimal-bayesian-recommendation-sets-and-myopically-optimal-choice-query-sets.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly: 'Pointer Networks', in: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2692–2700, 2015.

Xiaojun Wan and Jianmin Zhang: 'CTSUM: extracting more certain summaries for news articles', in: *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, pp. 787–796, 2014, Online: https://doi.org/10.1145/2600428.2609559.

Chenguang Wang, Laura Chiticariu, and Yunyao Li: 'Active Learning for Black-Box Semantic Role Labeling with Neural Factors', in: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI), August 19–25, 2017, Melbourne, Australia*, pp. 2908–2914, 2017a, Online: https://doi.org/10.24963/ijcai.2017/405.

Gaoang Wang, Jenq-Neng Hwang, Craig S. Rose, and Farron Wallace: 'Uncertainty sampling based active learning with diversity constraint by sparse selection', in: *19th IEEE International Workshop on Multimedia Signal Processing (MMSP), October 16–18, 2017, Luton, UK*, pp. 1–6, 2017b, Online: https://doi.org/10.1109/MMSP.2017.8122269.

Hao Wang, Naiyan Wang, and Dit-Yan Yeung: 'Collaborative Deep Learning for Recommender Systems', in: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pp. 1235–1244, 2015, Online: https://doi.org/10.1145/2783258.2783273.

Liangguo Wang, Jing Jiang, Hai Leong Chieu, Chen Hui Ong, Dandan Song, and Lejian Liao: 'Can Syntax Help? Improving an LSTM-based Sentence Compression Model for New Domains', in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 1385–1393, 2017c, Online: https://doi.org/10.18653/v1/P17-1127.

Lu Wang and Wang Ling: 'Neural Network-Based Abstract Generation for Opinions and Arguments', in: *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pp. 47–57, 2016.

Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu: 'Attention-Based Transactional Context Embedding for Next-Item Recommendation', in: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 2532–2539, 2018, Online: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16318.

Yaushian Wang and Hung-Yi Lee: 'Learning to Encode Text as Human-Readable Summaries using Generative Adversarial Networks', in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4187–4195, 01 2018.

Bartosz Wilczek and Claudia Blangetti: 'Live Blogging about Terrorist Attacks: The effects of competition and editorial strategy', *Digital Journalism* 6 (3): 344–368, 2018.

Kam-Fai Wong, Mingli Wu, and Wenjie Li: 'Extractive Summarization Using Supervised and Semi-Supervised Learning', in: *Proceedings of the 22nd International Conference on*

Computational Linguistics (COLING), Vol. 1, pp. 985–992, Manchester, UK, 2008a, Online: https://aclweb.org/anthology/C08-1124.

Kam-Fai Wong, Mingli Wu, and Wenjie Li: 'Extractive Summarization Using Supervised and Semi-Supervised Learning', in: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 985–992, Coling 2008 Organizing Committee, Manchester, UK, August 2008b, Online: https://www.aclweb.org/anthology/C08-1124.

Kristian Woodsend and Mirella Lapata: 'WikiSimple: Automatic Simplification of Wikipedia Articles', in: *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011, Online: http://www.aaai.org/ocs/index.php/AAAI/AAAI11/paper/view/3505.

Kristian Woodsend and Mirella Lapata: 'Multiple Aspect Summarization Using Integer Linear Programming', in: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*, pp. 233–243, Jeju Island, Korea, July 2012, Online: http://aclweb.org/anthology/D12-1022.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean: 'Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation', *arXiv* 1609.08144, 2016, Online: http://arxiv.org/abs/1609.08144.

Yuxiang Wu and Baotian Hu: 'Learning to Extract Coherent Summary via Deep Reinforcement Learning', in: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 5602–5609, 2018, Online: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16838.

Sander Wubben, Emiel Krahmer, Antal van den Bosch, and Suzan Verberne: 'Abstractive Compression of Captions with Attentive Recurrent Neural Networks', in: *Proceedings of the Ninth International Natural Language Generation Conference (INLG), September 5–8, 2016, Edinburgh, UK*, pp. 41–50, 2016, Online: http://aclweb.org/anthology/W/W16/W16-6608.pdf.

Zuobing Xu, Ram Akella, and Yi Zhang: 'Incorporating Diversity and Density in Active Learning for Relevance Feedback', in: *Advances in Information Retrieval, 29th European*

Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007, Proceedings, pp. 246–257, 2007, Online: https://doi.org/10.1007/978-3-540-71496-5_24.

Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen: 'Deep Matrix Factorization Models for Recommender Systems', in: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 3203–3209, 2017, Online: https://doi.org/10.24963/ijcai.2017/447.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir R. Radev: 'Graph-based Neural Multi-Document Summarization', in: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, pp. 452–462, 2017, Online: https://doi.org/10.18653/v1/K17-1045.

Wenpeng Yin and Yulong Pei: 'Optimizing Sentence Modeling and Selection for Document Summarization', in: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1383–1389, Buenos Aires, Argentina, 2015, Online: https://www.ijcai.org/Proceedings/15/Papers/199.pdf.

Chance York: 'Overloaded By the News: Effects of News Exposure and Enjoyment on Reporting Information Overload', *Communication Research Reports* 30 (4): 282–292, 2013.

Naitong Yu, Minlie Huang, Yuanyuan Shi, and Xiaoyan Zhu: 'Product Review Summarization by Exploiting Phrase Properties', in: *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pp. 1113–1124, 2016.

Naitong Yu, Jie Zhang, Minlie Huang, and Xiaoyan Zhu: 'An Operation Network for Abstractive Sentence Compression', in: *Proceedings of the 27th International Conference on Computational Linguistics (COLING), August 20-26, 2018, Santa Fe, NM, USA*, pp. 1065–1076, 2018, Online: https://aclanthology.info/papers/C18-1091/c18-1091.

Klaus Zechner: 'Automatic Summarization of Open-domain Multiparty Dialogues in Diverse Genres', *Journal of Computational Linguistics* 28 (4): 447–485, Dec 2002, Online: http://dx.doi.org/10.1162/089120102762671945.

Haiqin Zhang, Zheng Chen Wei-ying Ma, and Qingsheng Cai: 'A Study for Documents Summarization Based on Personal Annotation', in: *Proceedings of the HLT-NAACL 03 on Text Summarization Workshop*, pp. 41–48, 2003, Online: http://aclweb.org/anthology/W03-0506.

Kunpeng Zhang, Ramanathan Narayanan, and Alok N. Choudhary: 'Voice of the Customers: Mining Online Customer Reviews for Product Feature-based Ranking', in: *3rd Workshop*

on Online Social Networks, WOSN 2010, Boston, MA, USA, June 22, 2010, 2010, Online: https://www.usenix.org/conference/wosn-2010/voice-customers-mining-online-customer-reviews-product-feature-based-ranking.

Yi Zhang: *Bayesian Graphical Models for Adaptive Filtering*, Ph.D. thesis, Pittsburgh, PA, USA, 2005. AAI3191620.

Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma: 'Explicit factor models for explainable recommendation based on phrase-level sentiment analysis', in: *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 83–92, 2014.

Kaiqi Zhao, Gao Cong, Quan Yuan, and Kenny Q. Zhu: 'SAR: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews', in: *31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015*, pp. 675–686, 2015, Online: https://doi.org/10.1109/ICDE.2015.7113324.

Yang Zhao, Zhiyuan Luo, and Akiko Aizawa: 'A Language Model based Evaluator for Sentence Compression', in: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), July 15–20, 2018, Melbourne, Australia*, pp. 170–175, 2018, Online: https://aclanthology.info/papers/P18-2028/p18-2028.

Lei Zheng, Vahid Noroozi, and Philip S. Yu: 'Joint Deep Modeling of Users and Items Using Reviews for Recommendation', in: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pp. 425–434, ACM, Cambridge, United Kingdom, 2017, Online: http://doi.acm.org/10.1145/3018661.3018665.

Juming Zhou, Dong Wang, Yue Ding, and Litian Yin: 'SocialFM: A Social Recommender System with Factorization Machines', in: *Web-Age Information Management - 17th International Conference, WAIM 2016, Nanchang, China, June 3-5, 2016, Proceedings, Part I*, pp. 286–297, 2016, Online: https://doi.org/10.1007/978-3-319-39937-9_22.

Ming Zhou, Mirella Lapata, Furu Wei, Li Dong, Shaohan Huang, and Ke Xu: 'Learning to Generate Product Reviews from Attributes', in: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pp. 623–632, 2017.

Jingbo Zhu, Huizhen Wang, and Eduard H. Hovy: 'Learning a Stopping Criterion for Active Learning for Word Sense Disambiguation and Text Classification', in: *Third International Joint Conference on Natural Language Processing, IJCNLP 2008, Hyderabad, India, January 7-12, 2008*, pp. 366–372, 2008, Online: http://aclweb.org/anthology/I/I08/I08-1048.pdf.

Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen: 'Improving recommendation lists through topic diversification', in: *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005*, pp. 22–32, 2005, Online: https://doi.org/10.1145/1060745.1060754.

Markus Zopf: 'auto-hMDS: Automatic Construction of a Large Heterogeneous Multilingual Multi-Document Summarization Corpus', in: *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, pp. 3228–3233, Association for Computational Linguistics, Miyazaki, Japan, May 2018a.

Markus Zopf: 'Estimating Summary Quality with Pairwise Preferences', in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 1687–1696, 2018b, Online: https://aclanthology.info/papers/N18-1152/n18-1152.

Markus Zopf, Teresa Botschen, Tobias Falke, Benjamin Heinzerling, Ana Marasović , Todor Mihaylov, Avinesh P.V.S, Eneldo Loza Mencía, Johannes Fürnkranz, and Anette Frank: 'What's important in a text? An extensive evaluation of linguistic annotations for summarization', pp. 272–277, October 2018, Online: https://ieeexplore.ieee.org/document/8554853.

Markus Zopf, Maxime Peyrard, and Judith Eckle-Kohler: 'The Next Step for Multi-Document Summarization: A Heterogeneous Multi-Genre Corpus Built with a Novel Construction Approach', in: *Proceedings of the 26th International Conference on Computational Linguistics (COLING)*, pp. 1535–1545, Osaka, Japan, 2016, Online: https://aclweb.org/anthology/C16-1145.

Appendix

Notes on handling research data

According to the ”Guidelines on the handling of research data” of the Deutsche Forschungsgemeinschaft[¶], all the data and software related to this dissertation are archived and made pubicly available where possible.

The following research data has been made freely available:

- **Corpora**
  - The newly created corpus described in Section 3.2 is licensed under the Apache License 2.0 at https://github.com/UKPLab/lrec2018-live-blog-corpus.
- **Software**
  - The software required for the experiments described in Section 4.4 is available under the Apache License 2.0 license at https://github.com/UKPLab/acl2017-interactive_summarizer.
  - The software required for the experiments described in Section 5.3.1 is available under the Apache License 2.0 license at https://github.com/UKPLab/NAACL2019-interactiveCompression.
  - The application described in Section 4.7 is available under the Apache 2.0 license at https://github.com/UKPLab/vldb2018-sherlock http://sherlock.ukp.informatik.tu-darmstadt.de.
- **Research Results**
  - All publications related to this dissertation are available in the ACL Anthology (https://aclanthology.coli.uni-saarland.de/).
  - All research results are also documented in this dissertation itself, which is provided by the University and Regional Library Darmstadt.

[¶] http://dfg.de/download/pdf/foerderung/antragstellung/forschungsdaten/richtlinien_forschungsdaten.pdf

Further corpora described in this dissertation can not be made freely available for copyright reasons. According to the DFG Guidelines, these data and related software are archived internally using the infrastructure of the University and Regional Library Darmstadt, ensuring archiving for at least 10 years.

Publikationsverzeichnis des Verfassers

Avinesh P.V.S. and Christian M. Meyer (2017): Joint Optimization of User-desired Content in Multi-document Summaries by Learning from User Feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1353–1363, Vancouver, Canada.

Avinesh P.V.S., Maxime Peyrard and Christian M. Meyer (2018): Live Blog Corpus for Summarization. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC)*, pp. 3197–3203, Miyazaki, Japan.

Avinesh P.V.S., Benjamin Hättasch, Orkan Özyurt, Carsten Binnig and Christian M. Meyer (2018): Sherlock: A System for Interactive Summarization of Large Text Collections. In *Proceedings of the VLDB Endowment*, pp. 1902–1905, Rio de Janeiro, Brazil.

Andreas Hanselowski, **Avinesh P.V.S.**, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer and Iryna Gurevych (2018): A Retrospective Analysis of the Fake News Challenge Stance-Detection Task. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING), 2018*, pp. 1859–1874, Santa Fe, USA.

Markus Zopf, Teresa Botschen, Tobias Falke, Ana Marasovic, Todor Mihaylov, **Avinesh P.V.S.**, Eneldo Loza Mencía, Johannes Fürnkranz and Anette Frank (2018): What's Important in a Text? An Extensive Evaluation of Linguistic Annotations for Summarization. In *Proceedings of the 5th International Conference on Social Networks Analysis, Management and Security (SNAMS-18)*, pp. 272–277, Valencia, Spain.

Avinesh P.V.S. and Christian M. Meyer (2019): Data-efficient Neural Text Compression with Interactive Learning, In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, USA.

Avinesh P.V.S., Yongli Ren, Christian M. Meyer, Jeffrey Chan, Zhifeng Bao, Mark Sanderson (2019): J3R: Joint Multi-task Learning of Ratings and Review Summaries for Explainable Recommendation, In: *Proceedings of The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2019)*, Würzburg, Germany.

Wissenschaftlicher Werdegang des Verfassers[¶]

2004–2011	Computer Science Engineering Dual Degree International Institute of Information Techology Hyderabad (IIIT-H)
2008	Bachelors in Computer Science Engineering Referenten: Prof. Rajeev Sangal, Prof. Dipti Misra Sharma
2011	Master of Science in Computer Science Engineering Master-Thesis: "Transfer Grammar Engine and Automatic Learning of Reorder Rules in Machine Translation" Referenten: Prof. Rajeev Sangal, Prof. Dipti Misra Sharma
seit 2015	Wissenschaftlicher Mitarbeiter am Ubiquitous Knowledge Processing Lab Technische Universität Darmstadt

Ehrenwörtliche Erklärung[‡]

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades „Doktor-Ingenieur" mit dem Titel *„Information Preparation with the Human in the Loop"* selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Darmstadt, den 25. April 2019

Avinesh P.V.S.

[¶] Gemäß § 20 Abs. 3 der Promotionsordnung der Technischen Universität Darmstadt.

[‡] Gemäß § 9 Abs. 1 der Promotionsordnung der Technischen Universität Darmstadt.