SUMMARIZATION AND SENTIMENT ANALYSIS FOR UNDERSTANDING SOCIALLY-GENERATED CONTENT

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Lu Wang

February 2016

SUMMARIZATION AND SENTIMENT ANALYSIS FOR UNDERSTANDING SOCIALLY-GENERATED CONTENT

Lu Wang, Ph.D.

Cornell University 2016

During the past decades, we have witnessed the emergence of significant amounts of socially-generated content enabled by the widespread use of Internet, especially the social media websites. How to efficiently and effectively extract useful information and learn knowledge from the socially-generated content becomes a challenging task. Progress has been made in the area of natural language processing to help users understand and absorb knowledge from large volumes of text documents. This dissertation proposes broadly applicable natural language processing techniques to extract key information from massive amounts of heterogeneous textual data in response to users information queries and present it in a comprehensible way. Concretely, novel automatic summarization approaches are proposed to generate concise and informative responses from large amounts of texts to address users requests. We study textual data ranging from eloquent news articles written by professionals in traditional media, to massive user-generated content in popular social media, and to spontaneous conversations containing disfluency and interruptions. Furthermore, sentiment analysis methods are presented for studying the social interactions in online discussions. We target at discovering useful knowledge from informal text and thus obtaining a deeper understanding of socially-generated content.

BIOGRAPHICAL SKETCH

Lu Wang was born and grew up in Tianjin, China. She attended the experimental class in Yaohua High School from 2000 to 2005 in her hometown. After graduating from high school, she spent four years at Peking University in Beijing, and earned her bachelor degrees in Machine Intelligence and Economics in 2009. In the fall of 2009, she enrolled as a Ph.D. student in the Department of Computer Science at Cornell University. During the summer of 2012, she worked as an intern at IBM T.J. Watson Research Center in Yorktown Heights, New York. She also did an internship at Microsoft Research in Silicon Valley, California in the summer of 2013.

This thesis is dedicated to Yiye, who has always been supportive throughout the years. It is also dedicated to my parents, who raised me, inspired me, and taught me the first lessons in arithmetic, linear algebra, and programming.

ACKNOWLEDGEMENTS

First and foremost, I am immensely fortunate to have Professor Claire Cardie as my advisor, who has supported me tremendously throughout my Ph.D. life. Thanks to her invaluable advice and encouragement for guiding me to explore research challenges and thinking about scientific problems profoundly. She has been and will always be the person to look up to in my career.

I want to extend my gratitude to my committee members, Professor Johannes Gehrke and Professor Bruce Turnbull. They have provided numerous helpful suggestions, and without which this dissertation cannot be accomplished. Also, special thanks to Professor Lillian Lee, who is always generous to share her brilliant thoughts and suggestions for my work.

Furthermore, I am very grateful to my internship mentors: Hema Raghavan, Radu Florian, Vittorio Castelli (IBM T. J. Watson Research Center), Larry Heck, Dilek Hakkani-Tür, and Gokhan Tur (Microsoft Reseach). Under their supervision, I obtained exposure to research and development in an industrial environment, and acquired a significant amount of knowledge that directly contributes to this dissertation. There are also many other people that contributed to my enjoyable internship experience: Ding-Jung Han, Young-Suk Lee, Xiaoqiang Luo, Sameer Maskey, Salim Roukos, Todd Ward, Bowen Zhou (IBM), Gustavo Hernandez Abrego, Asli Celikyilmaz, and Malcolm Slaney (MSR).

I also must thank my collaborators and the previous and current members of Cornell NLP group: Carmen Banea, Xilun Chen, Yoonjung Choi, Cristian Danescu-Niculescu-Mizil, Lingjia Deng, John Hessel, Ozan Irsoy, Arzoo Katiyar, Lillian Lee, Moontae Lee, Parvaz Mahdabi, Galen Marchetti, Rada Mihalcea, Myle Ott, Jon Park, Karthik Raman, Vikram Rao, Dinesh Puranam, Xanda Schofield, Adith Swaminathan, Chenhao Tan, Janyce Wiebe, Xiaoan Yan, Bishan Yang, Ainur Yessenalina. I enjoyed the fun

discussions with these brilliant minds, which inspired many interesting research ideas.

Finally, I would like to thank the most important ones in my life: my mother, my father, and my husband, Yiye Ruan. I can never thank you enough for all the support and love you give me, and that is what makes me, beyond this dissertation.

The work in this dissertation was supported in part by the National Science Foundation under Grants IIS-0535099, IIS-0968450, IIS-1111176, and IIS-1314778, DARPA DEFT Grant FA8750-13-2-0015, a gift from Google, and a gift from Boeing.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . vii
List of Tables . . . x
List of Figures . . . xii

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Contributions . . . 5
1.2.1 Automatic Text Summarization . . . 5
1.2.2 Understanding Online Social Interactions . . . 8

2 Background **10**
2.1 Text Summarization: A Brief History . . . 10
2.1.1 Generic vs. Focused Summarization . . . 10
2.1.2 Extractive vs. Abstractive Summarization . . . 12
2.1.3 What Makes a Good Summary? . . . 17
2.1.4 Summarization Evaluation: A Harder Problem? . . . 18
2.2 Genre- and Domain-Specific Summarization . . . 21
2.2.1 Focused Summarization for Spoken Meetings . . . 21
2.2.2 Summarization for User-Generated Content . . . 23
2.2.3 Timeline Generation . . . 25
2.3 Sentiment Analysis for User-Generated Content . . . 26
2.3.1 Why Do We Care About Sentiment? . . . 27
2.3.2 Agreement and Disagreement Detection in Online Social Interactions . . . 27

3 Meeting Summarization: Beyond Utterance Extraction **30**
3.1 Introduction . . . 30
3.2 Token-Level Representation via Unsupervised Topic Modeling . . . 36
3.2.1 Summarization Frameworks . . . 37
3.2.2 Topic Models . . . 40
3.2.3 Experimental Setup . . . 42
3.2.4 Results . . . 45
3.3 Structured Representation via Unsupervised Relation Extraction . . . 51
3.3.1 Focused Summarization as Relation Extraction . . . 52
3.3.2 The Relation Extraction Model . . . 53
3.3.3 Parameter Estimation and Inference via Posterior Regularization 54
3.3.4 Features . . . 57
3.3.5 Experimental Setup . . . 59
3.3.6 Results . . . 62
3.4 Conclusion . . . 64

4 Abstract Generation for Multi-Party Meetings **66**
4.1 Introduction . . . 66
4.2 The Framework . . . 69
4.3 Content Selection . . . 70
4.4 Surface Realization . . . 72
4.4.1 Template Extraction . . . 73
4.4.2 Template Filling . . . 75
4.4.3 Post-Selection: Redundancy Handling. . . . 78
4.5 Experimental Setup . . . 78
4.6 Results . . . 80
4.7 Conclusion . . . 83

5 Sentence Compression to Multi-Document Summarization **89**
5.1 Introduction . . . 89
5.2 The Framework . . . 92
5.3 Sentence Compression . . . 94
5.3.1 Rule-based Compression . . . 95
5.3.2 Sequence-based Compression . . . 95
5.3.3 Tree-based Compression . . . 97
5.4 Experimental Setup . . . 104
5.5 Results . . . 105
5.5.1 Automatic Evaluation . . . 105
5.5.2 Human Evaluation . . . 107
5.5.3 Sentence Compression Evaluation . . . 108
5.6 Conclusion . . . 110

6 Summarizing Opinion from Social Media **111**
6.1 Introduction . . . 111
6.2 Submodular Opinion Summarization . . . 114
6.2.1 Relevance Function . . . 115
6.2.2 Coverage Functions . . . 116
6.2.3 Dispersion Function . . . 118
6.2.4 Full Objective Function . . . 120
6.2.5 Summary Generation via Greedy Algorithm . . . 120
6.3 Experimental Setup . . . 121
6.3.1 Opinion Question Identification . . . 121
6.3.2 Datasets . . . 121
6.3.3 Comparisons . . . 122
6.4 Results . . . 123
6.4.1 Evaluating the Ranker . . . 123
6.4.2 Community QA Summarization . . . 124
6.4.3 Blog Summarization . . . 127
6.4.4 Further Discussion . . . 129
6.5 Conclusion . . . 131

7 Socially-Informed Timeline Generation **132**
7.1 Introduction . . . 132
7.2 Data Collection and Preprocessing . . . 134
7.3 Joint Learning for Importance Scoring . . . 135
7.4 Timeline Generation . . . 139
7.4.1 Entity-Centered Event Threading . . . 140
7.4.2 Summary Quality Measurement . . . 141
7.4.3 Connectivity Measurement . . . 142
7.4.4 An Alternating Optimization Algorithm . . . 143
7.5 Experimental Results . . . 145
7.5.1 Evaluation of SENTENCE and COMMENT Importance Scorers . 145
7.5.2 Leveraging User Comments . . . 146
7.5.3 Evaluating Socially-Informed Timelines . . . 147
7.5.4 Human Evaluation of Event Threading . . . 149
7.6 Conclusion . . . 150

8 Sentiment Analysis for Online Social Interaction **154**
8.1 Introduction . . . 154
8.2 Agreement and Disagreement Identification in Online Discussions . . . 159
8.2.1 The Model . . . 159
8.2.2 Online Discussion Sentiment Lexicon Construction . . . 163
8.2.3 Experimental Setup . . . 165
8.2.4 Results . . . 167
8.3 Online Dispute Detection with Sentiment Analysis Approach . . . 171
8.3.1 Data Construction: A Dispute Corpus . . . 171
8.3.2 Sentence-level Sentiment Prediction . . . 173
8.3.3 Online Dispute Detection . . . 175
8.4 Conclusion . . . 180

9 Conclusion and Future Horizons **182**
9.1 Conclusion . . . 182
9.2 Future Horizons . . . 184

LIST OF TABLES

3.1 Experimental results for token-level focused meeting summariztion systems with true clusterings of dialogue acts as input. 49
3.2 Experimental results for token-level focused meeting summariztion systems with system generated clusterings of dialogue acts as input. . . 50
3.3 Sample summaries generated by topic modeling-based systems. 51
3.4 Features for Decision Cue and Decision Content relation extraction. . . 58
3.5 Sample Decision Cue relation instances discovered by the unsupervised relation extraction model. 59
3.6 Additional features for Decision Content relation extraction model. . . 59
3.7 Experimental results on decision summarization for spoken meeting with perfect clusterings of Decision-Related Dialogue Act. 62
3.8 Experimental results on decision summarization for spoken meeting with system generated clusterings of Decision-Related Dialogue Act. . 64
3.9 Sample system outputs by different meeting summarization methods. . 65

4.1 Features used for content selection in the meeting abstract generation system. 72
4.2 Features for abstracts ranking in the meeting abstract generation system. 77
4.3 Domain adaptation evaluation for the meeting abstract generation system. 82
4.4 Human evaluation results of fluency and semantic correctness for the generated abstracts. 83

5.1 Sentence ranking features for query-focused multi-document summarization system. 93
5.2 Linguistically-motivated rules for sentence compression. 95
5.3 Token-level features for sequence-based sentence compression. 96
5.4 Constituent-level features for tree-based sentence compression. 102
5.5 Automatic evaluation results on query-focused multi-document summarization systems. 106
5.6 Human evaluation results on query-focused multi-document summarization systems. 107
5.7 Evaluation results on sentence compression. 110

6.1 Features used for candidate ranking in the opinion summarization system. 116
6.2 Experimental results for best answer prediction by average precision and mean reciprocal rank (MRR). 123
6.3 Summaries evaluated by Jensen-Shannon divergence (JSD) on Yahoo Answer dataset. 125
6.4 Value addition of each component in the objective function for generating opinion summaries. 125
6.5 Human evaluation on opinion summarization for Yahoo! Answer Data. 126
6.6 Automatic evaluation on opinion summarization for TAC'08 dataset. . 129
6.7 Human evaluation on opinion summarization for TAC'08 dataset. . . . 129

6.8 Effect of different dispersion functions, content coverage, and dissimilarity metrics on our opinion summarization system for Yahoo! data. . 130
6.9 Effect of different dispersion functions, content coverage, and dissimilarity metrics on our system for TAC'08 dataset. 130

7.1 Statistics on the four event datasets for socially-informed timeline generation. 135
7.2 Features used for sentence importance scoring for our socially-informed timeline generation system. 137
7.3 Features used for comment importance scoring for our socially-informed timeline generation system. 138
7.4 Automatic evaluation for timeline generation systems by using ROUGE. 147
7.5 Human evaluation results on the comment portion of socially-informed timelines. 148
7.6 Human evaluation on the informativeness of answers written after reading timelines. 150

8.1 Features used in sentiment prediction for online discussions. 162
8.2 Example terms and relations from the online discussion lexicon. 163
8.3 F1 scores for agreement and disagreement detection on Wikipedia talk pages. 169
8.4 F1 scores for agreement and disagreement detection on online debate (IAC) 170
8.5 Results on Wikipedia talk page (AAWD) and online debate (IAC) with different feature sets. 171
8.6 Relevant features for agreement and disagreement detection by χ^2 test on Wikipedia Talk pages and online debates. 171
8.7 Subcategory for disputes with corresponding tags. 172
8.8 Features used in sentence-level sentiment prediction. 174
8.9 F1 scores for positive and negative alignment on Wikipedia Talk pages (AAWD). 177
8.10 Dispute detection results on Wikipedia Talk pages. 179
8.11 Dispute detection results with different feature sets by SVM with RBF kernel. 179

LIST OF FIGURES

3.1 A clip of a meeting from the AMI meeting corpus [Carletta et al., 2005]. 32
3.2 Clip from the AMI meeting corpus [Carletta et al., 2005]. 34
3.3 Experimental results on focused meeting summarization with different topic models. 45
3.4 Comparisons between token-level meeting summarization systems and utterance-level meeting summarization systems. 46
3.5 Experimental results on leveraging context information for focused meeting summarization. 47
3.6 Comparisons between token-level meeting summarization systems and utterance-level meeting summarization systems on leveraging context information. 48
3.7 Graphical model representation for the relation learning model. 54

4.1 Clips from the AMI meeting corpus [Mccowan et al., 2005]. 67
4.2 The meeting abstract generation framework. 84
4.3 Example of template extraction by Multiple-Sequence Alignment for problem abstracts from AMI. 85
4.4 Content selection evaluation results for the meeting abstract generation system by using ROUGE-SU4. 86
4.5 Full meeting abstract generation system evaluation by using BLEU. . . 87
4.6 Sample decision and problem summaries generated by various systems. 88

5.1 Diagram of tree-based compression. 98
5.2 Example of beam search decoding for tree-based sentence compression. 100
5.3 Sample summary generated by our sentence compression based query-focused MDS system. 109

6.1 Example discussion on Yahoo! Answers. 112
6.2 Sample opinion summaries for different systems. 128

7.1 A sample timeline on Ukraine crisis with social context from user comments. 151
7.2 An example on computing the connectivity between an article summary and a comment summary via best matching in bipartite graph. 152
7.3 Evaluation of sentence and comment ranking by using normalized discounted cumulative gain at top 3 returned results (NDCG@3). 153
7.4 An example output of our socially-informed timeline generation system. 153

8.1 Example discussion from wikipedia talk page for article. 155
8.2 A sample discussion from Wikipedia Talk page that contains dispute. . 158
8.3 Sentiment flow visualization for discussions with dispute. 181

CHAPTER 1

INTRODUCTION

1.1 Motivation

With the development of the Internet, people can obtain and share information almost instantly from a wide array of sources, for example, online news outlets and fast-growing social networks. At the same time, it has become increasingly convenient for Internet users to generate an immense amount of Web data, in various forms such as text, images, or video. Text, as a specific type of data, has long been integral to knowledge sharing and discovery. Over the past few decades, the explosive growth of textual data far outpaces human beings' speed of understanding its content. Indeed, we have seen the emergence of new types of textual data that reflect social interactions in online settings. The production of this socially-generated content is accelerated by the wide adoption of social media sites, such as Facebook, Twitter, Yahoo! Answers, and Reddit.

We are therefore facing an inevitable and challenging problem: information overload. Fortunately, the area of information retrieval [Salton and McGill, 1986] has achieved great success in the last decades. As one of its most significant applications, search engines have enabled users to retrieve information from digital collections by providing a ranked list of documents or web pages, given a user-specified query. However, even the most sophisticated search engines empowered by advanced information retrieval techniques lack the ability to synthesize information from multiple sources and present users with a concise yet informative response. It still requires a significant amount of time for users to sift through the retrieved documents to absorb, interpret, and organize the information. Redundancy among the web pages further hinders efficient information seeking.

To help users absorb knowledge efficiently and effectively, this dissertation proposes *the use of natural language processing (NLP) techniques to extract key information from text in response to users' requests and present it in a comprehensible way*. As aforementioned, the rapid growth of text data comes with pressing challenges for designing robust and scalable NLP models and algorithms. These challenges not only arise from the data itself, but also involve the users – the information seekers.

- **Challenge I: Massive Amount of Textual Data.** The booming growth of the Internet makes it possible for the creation and exchange of large amounts of textual data. For example, on a daily basis, millions of blogs are written, hundreds of millions of tweets are sent out, and billions of queries are made on search engines.[1] For English Wikipedia alone, there are almost 5 million articles, and more than 1,000 new articles created every day.[2] Therefore, scalable and robust algorithms are needed to extract the specific information that users care about from this mass of data.

- **Challenge II: The Pervasiveness of Heterogeneity.** There also arises textual data of disparate types and genres, each differing from others in various respects. (1) Firstly and most obviously, the texts cover *different topics*, from social issues to entertainment. (2) The different sources of textual data can also *each serve a distinct social purpose*. Posts to online discussion forums, for example, allow users to express their opinions; and customers can use online review services, like Yelp or TripAdvisor, to share their experiences. Moreover, Internet users can also help each other answer questions through community question answering services. Lastly, people not only communicate on social media, but also collaborate

[1] Source: `http://www.internetlivestats.com/twitter-statistics/`. Accessed on June 28, 2015.

[2] The statistics are from `https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia` in June of 2015.

to construct knowledge, e.g. Wikipedia – the high quality encyclopedic resource. (3) Not surprisingly, textual data is presented in *different formats and using different writing styles*. For instance, well-constructed news articles are used to report recent news comprehensively and are written in a professional style. In contrast, one-sentence microblogs are usually used to break the news, share personal experience, or express opinions, and they are penned using very informal language. (4) Furthermore, data from different media also *varies in terms of volume and time scale*. For example, instant messages offer us real-time interactions, while emails account for communications over a longer time span. Taken together, these factors establish a need for computational methods to deal with many different genres of text. They also imply that there is no one-size-fits-all solution for text modeling and analysis.

- **Challenge III: The Emergence of Informal Text.** Though success has been made for a wide range of NLP tasks, including syntactic analysis, named-entity recognition, relation extraction, and question answering, the progress is still limited to analyzing texts written in a fairly formal style and undergone editing, such as news articles. However, textual data captured in social media is difficult to handle with NLP tools that have instead been optimized for edited text. Improving the automatic analysis of informal text would require a number of changes including (1) newly annotated datasets to train NLP models, and (2) novel algorithms to capture the genre-specific properties.

- **Challenge IV: Diverse User Information Needs.** Another challenge comes from the users, who exhibit diverse information needs. Even for the same data source, users might seek different types of information. For instance, people may subscribe to the same news media out of different personal interests, and thus expect to get different information from it. There are several reasons why this is chal-

lenging and why efforts should be made to take user factors into consideration. (1) Firstly, users can adjust and change their information need from time to time. At the onset of an event, for example, readers might want to see a general description of what has happened, but after that, they would likely look for more details or be interested in how other people are reacting to the event. (2) Even when different users look for the same type of information, the disparity in their knowledge levels demands that the information be presented in different ways. For instance, when presenting the cause of a disease for a domain expert, relevant medical terms are appropriate to include in the explanation. However, simpler language is better when providing explanations to common patients. (3) Users' information needs can also be reflected in their behavior patterns for communication or search; these can be learned to further improve user experience. Web search and browsing is a good example. Users who are familiar with a certain topic can successfully find the right answer after several clicks. However, those who do not really know what to search for may end up with tens of page views. Query intentions can be learned from these search behaviors, and thus help build a better search engine. Given the above rationales, we believe it is necessary to analyze and model the users' request to meet their information needs.

The research described in this dissertation is motivated by the above challenges. We will present broadly applicable NLP techniques that efficiently analyze large amounts of textual data from various domains and of different genres to produce responses that meet users' diverse information needs.

We first explore novel automatic text summarization approaches to generate concise and informative responses from massive amounts of heterogeneous texts to address users' requests. These texts range from eloquent news articles written by profession-

als, to massive user comments and blogs with conflicting opinions, and to spontaneous spoken meetings containing disfluency and interruptions. We then investigate sentiment analysis methods to study the social dynamics in online discussions. This part of the work aims to extract useful knowledge from informal text and thus obtain a deeper understanding of socially-generated content. The corresponding contributions are described in more detail in the next section.

1.2 Contributions

The contributions of this dissertation fall into two areas – text summarization and sentiment analysis.

1.2.1 Automatic Text Summarization

We have seen considerable progress made for text summarization over the years, and solutions to certain tasks have already materialized into product features such as Google's Answer Box. Despite all these developments, existing systems are still limited in that they are largely *extractive* [Goldstein et al., 1999] — the summary is comprised of important sentences drawn verbatim from the original documents. At the same time, existing systems are typically designed for highly edited text from well-studied genres, like news reports or academic articles. Part of our research makes steps towards generating *abstractive summaries* [Gerani et al., 2014] — we first try to understand the input and then generate the summary "from scratch", including sentences or phrases that did not appear in the original document. We further target unedited text of a much more informal nature, such as spoken meeting transcripts. This work will be presented in

Chapters 4 and 5.

Part of our summarization work makes contributions in the area of *focused summarization* [Conroy et al., 2006a]. A focused summarization system aims to generate summaries with respect to a particular aspect of the information of interest or in response to a user-specified query. We argue that focused summarization is a better fit for addressing users' information needs compared to generic summarization, where system-generated summaries are not customized for distinct information requests. This work will be described in Chapters 3, 4, 5, and 6.

Moreover, the reading experience itself can determine how effectively users become informed [Stanovich, 1986]. Existing summarization systems usually present a summary as one paragraph [Radev and McKeown, 1998], and the structural relations among different pieces of text, such as temporal sequence, are thus missing. Making connections among relevant texts can help readers acquire knowledge and absorb information efficiently [Shahaf and Guestrin, 2010]. Along these lines, we will describe a new timeline generation framework for complex events in Chapter 7.

In general, our contributions on automatic text summarization are three-fold:

Generating high quality focused abstractive summaries for informal noisy text. We propose a complete and fully automatic domain-independent abstract generation framework for focused meeting summarization. We approach the content selection step as a relation extraction task. We then apply Multiple-Sequence Alignment to induce abstract generation templates that can be used for different meeting topics. An Overgenerate-and-Rank strategy is utilized to produce and rank candidate abstracts. These approaches are described in detail in Chapters 3 and 4, where methodology, datasets, and experiments are explained.

Addressing open-ended information requests. We first consider the task of query-focused multi-document summarization on news articles. We focus on developing sentence compression-based approaches to further condense text by removing secondary information from lengthy sentences while retaining the salient content that users seek for. This system and the corresponding experiments are introduced in Chapter 5. We then investigate the task of opinion summarization on community questions answering and blogs given a user's query. This is implemented by a submodular function-based framework. Within this framework, relevance ordering produced by a statistical ranker, and information coverage with respect to topic distribution and diverse viewpoints are both encoded as submodular functions. Dispersion functions are utilized to minimize the redundancy. More information on the techniques and experiments are described in Chapter 6.

Proposing a new framework for creating summaries that enrich users' reading experience. We present a socially-informed timeline generation system that jointly generates a news article summary and a user comment summary for each day of an ongoing complex event. We maximize topic cohesion between the article and comment summaries while preserving their ability to reflect important concepts and subevents, adequate coverage of mentioned topics, and continuity of the timeline as it is updated with new materials each day. We design a novel alternating optimizing algorithm that allows the generation of a high quality article summary and comment summary via mutual reinforcement. We will provide more details on the frameworks in Chapter 7, including the algorithms, corpora, and evaluations.

1.2.2 Understanding Online Social Interactions

The last part of this dissertation is also connected to the research area of Computational Social Science [Cioffi-Revilla, 2014], which studies how people interact with each other. Computational Social Science is a rising interdisciplinary field drawing attentions from the areas of computer science, mathematics, social science, political science, and others.

Analyzing online social interactions has attracted a significant amount of work from various research areas in computer science, such as natural language processing [O'Connor et al., 2010, Eisenstein et al., 2011], data mining [Leskovec et al., 2009, Romero et al., 2011, Satuluri et al., 2011], and machine learning [Chang et al., 2009]. We are interested in studying the language that people use in online social interactions as a mechanism for understanding the underlying social dynamics. Firstly, the online communication setting provides many examples of conversations that can bring insights into the ways people communicate with others of different relationships and for different purposes. Automatic methods and computational models are thus needed to understand the social dynamics. Furthermore, it becomes effortless to share and exchange opinions on the Internet. This, however, makes it difficult to collect comprehensive viewpoints and reason from this rich data resource. It is necessary to automate the sentiment prediction and opinion extraction process. Finally, the development of Internet also expedited the production of collaboratively generated content, such as the millions of articles on Wikipedia created in the past decade. Effective online collaboration usually benefits the creation of the content. In order to facilitate fruitful collaboration, automatic conflict and dispute detection components are necessary.

Therefore, we have two main contributions on utilizing sentiment analysis techniques to study online social dynamics:

Sentiment prediction in online discussions. We study the problem of detecting agreement and disagreement sentences in online conversations, such as the ones on Wikipedia talk pages or debate forums. An isotonic Conditional Random Fields based sequential model is proposed to make predictions at the sentence- or segment-level. We automatically construct a socially-tuned sentiment lexicon that is bootstrapped from existing general-purpose sentiment lexicons to further improve the performance. This work will be presented in Section 8.2.

Online dispute detection. We utilize a sentiment classifier to investigate the task of online dispute detection. We first identify the sequence of sentence-level sentiments expressed during a discussion, and then use them as features in a classifier that predicts the dispute/non-dispute label for the discussion as a whole. Detailed algorithms and experiments along with the newly constructed dataset from Wikipedia talk pages are introduced in Section 8.3.

CHAPTER 2

BACKGROUND

2.1 Text Summarization: A Brief History

In this section, we will start by introducing different types of text summarization and the corresponding related work in each area. Early work in summarization focused on *single-document summarization* [Goldstein et al., 1999], where the goal is to construct a summary for one input document, such as a news article, an academic talk, or a weblog. Later on, with the booming growth of text data, there arose a great need for *multi-document summarization* (MDS) systems [Radev and McKeown, 1998]. In this dissertation, our work on meeting summarization falls into the realm of single document summarization. For the rest of the summarization work in this dissertation, however, we target at modeling and understanding a large number of web documents, and that part is considered as multi-document summarization.

2.1.1 Generic vs. Focused Summarization

In general, there are two major types of summarization systems based on whether the system considers user information. *Generic summarization* assumes that the audience that reads the summary is a general one. In this setting, a generic summarization system determines the appropriateness of including a phrase or a sentence into the summary only based on the information contained in the input documents. Another type of summarization – *focused summarization*, targets at generating summaries for specific information of interest, especially for the information requested by users. Specifically, a *query-focused summarization* system usually takes a question asked by a user, and

then generates a summary with respect to the query, ignoring all other content from the original document(s). Our summarization work all lies in the context of focused summarization.

Much of the previous work has been in the context of generic summarization though. It mostly focuses on determining important content to be included in the summary based on word frequencies [Luhn, 1958, Lin and Hovy, 2000, Nenkova et al., 2006], sentence centrality [Carbonell and Goldstein, 1998, Erkan and Radev, 2004], or information coverage [Haghighi and Vanderwende, 2009]. Between 2001 and 2004, the Document Understanding Conferences (DUC) organized quantitative evaluations of single document and multi-document summarization systems for generic summarization. After the DUC datasets that consist of document clusters and human written abstracts were made available, there emerged a huge number of research efforts with the goal of developing and evaluating generic multi-document summarization [Nenkova et al., 2006, Wang et al., 2008, Haghighi and Vanderwende, 2009, Lin and Bilmes, 2010, Lin and Hovy, 2003].

On the other hand, Jones [1998] has pointed out that "context factors" are very important for summarization, and "the idea of a general-purpose summary is manifestly an *ignis fatuus*". We believe generic summarization systems have many practical usages, but we also argue that query-focused summarization is more realistic in many scenarios. As stated earlier, users with different backgrounds seek different types of information. Thus, focused summarization systems need to consider the information from both user queries and the documents to be summarized. If designed properly, query-focused summarization can potentially facilitate other applications, such as generating snippets for web search results [Turpin et al., 2007], or supporting an open-ended question answering system [Wang et al., 2014].

Since 2005, DUC conferences shifted their evaluations to query-focused multi-

document summarization, which spurred more research in this area. This direction has also been adopted by the succeeding Text Analysis Conferences (TAC). A wide range of methods have been investigated for query-focused summarization. Though many of them are extended from generic multi-document summarization systems, new systems usually include components on estimating the query relevance of the generated summary. For instance, the Maximal Marginal Relevance based framework [Carbonell and Goldstein, 1998] presents a criterion for re-ranking text snippets according to their query relevance and redundancy with regard to the existing summary. In a similar spirit, submodular functions are also proposed to balance between the query relevance and diversity of the summary [Lin and Bilmes, 2011]. Those approaches measure the relevance of the summaries through TF-IDF similarity [Lin and Bilmes, 2011, Dasgupta et al., 2013]. Query relevance can also be evaluated by language modeling [Conroy et al., 2006a] or by learning a Bayesian model over queries and documents [Daumé and Marcu, 2006]. From users' point of view, one can modify one's queries based on the previous summaries generated from the system. This idea has been studied as a query-chain summarization task [Baumel et al., 2014], where a series of relevant queries are considered, and an update summary is constructed for each query in the chain.

2.1.2 Extractive vs. Abstractive Summarization

For the past decades, the most prominent multi-document summarization approaches have been *extractive summarization* methods, where sentences from the original documents are selected for inclusion in the final summary. Extractive methods have been popular mainly because they are relatively simple to construct, since the the problem can be converted to a sentence selection task and the output summary does not suffer from ungrammaticality. Various methods have been employed for estimating the

informativeness or importance of the input sentences. For unsupervised methods, sentence importance can be measured by calculating topic signature words [Lin and Hovy, 2000, Conroy et al., 2006b], estimating text centrality within a graph-based model [Otterbacher et al., 2005], or using a Bayesian model with sophisticated inference [Daumé and Marcu, 2006]. For example, Davis et al. [2012] first learn the term weights by Latent Semantic Analysis, and then greedily select sentences that cover the maximum combined weights. Lin and Bilmes [2011] use a class of carefully designed submodular functions to reward the diversity of the summaries and select sentences greedily. Supervised approaches often employ discriminative learning to rank sentences [Fuentes et al., 2007]. Sipos et al. [2012] extends the submodular function-based framework of Lin and Bilmes [2011] by using a large-margin-based training to learn the weights of ngrams. More recently, recursive neural networks have been investigated to rank the sentences based on their importance [Cao et al., 2015].

Though progress has been made in extractive summarization, one of the problems that extract-based approaches suffer from is that they unavoidably include secondary or redundant information. More importantly, it is still far from the way humans write summaries. When people write summaries, they tend to abstract the content and seldom use entire sentences taken verbatim from the original documents. If we compare the human summaries with the input documents, we can observe several operations on how humans use and modify the input content: sentence compression, information fusion, paraphrasing, and generation. Therefore, summarization research has moved towards the area of *abstractive summarization*. Abstract-based methods are often designed to approximate how human construct summaries. Here we describe a few promising directions that have been pursued in this area.

Sentence Compression. Human writers sometimes remove redundant or irrelevant information within lengthy sentences when they construct summaries. Sentence compression has been recently investigated to produce a compact and grammatical version of a sentence while preserving salient information. Our research described in Chapter 5 is inspired by probabilistic sentence compression approaches, such as the noisy-channel model [Knight and Marcu, 2000, Turner and Charniak, 2005], and its extensions via synchronous context-free grammars (SCFG) [Aho and Ullman, 1969, Lewis and Stearns, 1968] for robust probability estimation [Galley and McKeown, 2007]. Rather than attempt to derive a new parse tree like Knight and Marcu [2000] and Galley and McKeown [2007], we learn to safely remove a set of constituents in our parse tree-based compression model while preserving grammatical structure and essential content. Sentence-level compression has also been examined via a discriminative model [McDonald, 2006]. Discourse information has been found useful and is incorporated into the compression process by integer linear programming [Clarke and Lapata, 2008].

Our work is more related to the less investigated area of sentence compression as applied to document summarization. Zajic et al. [2006] tackle the query-focused multi-document summarization (MDS) problem using a compress-first strategy: they develop heuristics to generate multiple alternative compressions of all sentences in the original document as candidates for extraction. This approach, however, does not outperform some extraction-based approaches. A similar idea has been studied for MDS [Lin, 2003, Gillick and Favre, 2009], but limited improvement is observed over extractive baselines with simple compression rules. Finally, although learning-based compression methods are promising [Martins and Smith, 2009, Berg-Kirkpatrick et al., 2011, Li et al., 2013a, 2014], it is unclear how well they handle issues of redundancy. For instance, Almeida and Martins [2013] present a dual decomposition framework to extract and compress sentences simultaneously. The parameters of extraction and compression models are

acquired by their multi-task learning approach.

Sentence Fusion. In addition to compression, sentence fusion is another task designed to generate non-extractive summaries. The goal of sentence fusion is to merge multiple sentences into one by removing duplicate information while reserving fragments that are different. The core questions to study for fusing multiple sentences are to align the common information and determine which part of the sentences are to be retained. Barzilay and McKeown [2005] first identify the sentence with the most essential information and augment it with information from other sentences. In order to better guide the alignment and merging processes, supervised learning based methods have been investigated [Elsner and Santhanam, 2011, Thadani and McKeown, 2013], where the dependency trees of the input sentences are merged using Integer Linear Programming. Cheung and Penn [2014] later expand the sentence fusion process with external resources beyond the input sentences. Banerjee et al. [2015] first cluster similar sentences into different groups. Each group of sentences is aligned into a graph and the best path is identified as the output sentence with an Integer Linear Programming algorithm.

Abstractive Summarization. Generating abstracts for a set of relevant documents is a more challenging task than all the techniques mentioned above can handle, because it requires both techniques for language understanding and generation. Many factors need to be considered for building a complete abstract generation system, even for a specific domain or application. In addition to informativeness and non-redundancy, ensuring a good reading experience further requires the summaries to be grammatical, coherent, and semantically correct. Most previous work focuses on addressing one or two of these issues and ignoring the others.

Existing work in abstractive summary generation is limited to specific domains,

where fixed templates or rules are manually crafted for generating the sentences. For example, abstract-based approaches have been studied for product reviews that tend to have significant amount of redundant information. Ganesan et al. [2010] utilize a graph-based algorithm to merge reviews that have similar textual content. Though their approach is able to remove some redundancy for the cases where the reviews contain the same textual information, there is no guarantee for the output summary to be grammatical. To ensure the grammaticality, Gerani et al. [2014] design a set of sentence realization templates for summary sentences that serve different discourse functions. The discourse function of each sentence is determined by the aspect rhetorical relation graph constructed from reviews for each product. Instead of generating a summary consisting of multiple sentences, Pighin et al. [2014] focus on only generating a headline for each news article sentence. They first learn the event templates from a large number of news articles with a memory-based pattern extraction model, and then fill the entities into appropriate templates to form the headline.

Our work is also broadly related to expert system-based language generation [Reiter and Dale, 2000] and concept-to-text generation tasks [Angeli et al., 2010, Konstas and Lapata, 2012], where the generation process is decomposed into content selection (or text planning) and surface realization. For instance, Angeli et al. [2010] learn from structured database records and parallel textual descriptions. They generate texts based on a series of decisions made to select the records, fields, and proper templates for rendering. Those techniques that are tailored to specific domains (e.g. weather forecasts or sportcastings) cannot be directly applied to data of new domains or genres, as their input is well-structured and the templates learned are domain-specific.

2.1.3 What Makes a Good Summary?

Clearly, there are lots of good properties embodied in a high quality summary, such as high informativeness, clarity, coherence, and low redundancy. Here we stress the following facets: informativeness, diversity, and coherence, which will be investigated in detail in this dissertation.

Informativeness is one of the most important properties to be modeled in a summary and is usually estimated at the sentence-level. Under query-focused summarization, relevancy is always measured along with sentence importance. As we have described for sentence extraction techniques in the previous section, importance can be calculated by counting topic signature words [Lin and Hovy, 2000, Conroy et al., 2006b], estimating text centrality [Otterbacher et al., 2005], or using a Bayesian model [Daumé and Marcu, 2006]. Discriminative learning is also exploited for ranking sentences [Fuentes et al., 2007].

Encouraging the information **diversity** of a summary is necessary because summarization systems generally produces summaries within a length constraint. An equivalent task is to minimize redundancy. Previous work concentrates on improving the estimation of sentence relevance, while summary diversity is only implicitly modeled, for example, by downweighting the importance scores of words that are already selected. An exception is Maximal Marginal Relevance (MMR) [Carbonell and Goldstein, 1998], an approach that selects sentences in a greedy fashion that trades-off relevance and summary redundancy. This technique is however sub-optimal as each sentence is selected or discarded based only on the ones selected previously. Proposed ILP formulations of MMR [McDonald, 2007] that overcome this limitation are not scalable. Encouraging diversity within a summary has recently been addressed through submodular functions, which have been applied for multi-document summarization in newswire [Lin and

Bilmes, 2011, Sipos et al., 2012], and comment summarization [Dasgupta et al., 2013].

Given that informativeness and diversity are two important factors for summarization, some work treats them as a trade-off and utilizes optimization process to find a balance between the two. Integer Linear Programming (ILP) is a popular approach. For example, Gillick and Favre [2009] design an objective function as the sum of weighted concepts in the summary, and use the ILP framework to select sentences for inclusion in the summary. Concepts are usually represented as bigrams. Built on this work, Li et al. [2013b] further learned the frequency of each concept with supervised training.

Coherence is another important factor for all types of writing and also for generating high quality summary. Previously, optimizing the coherence of a summary has been approached as a sentence ordering task [Lapata, 2003, Barzilay and Lapata, 2005], which follows the step of sentence selection. Barzilay and Lapata [2005] present an entity-based representation of discourse to support their local coherence assessment model. Christensen et al. [2013] try to solve the sentence selection and ordering tasks simultaneously, where they use an optimization framework to jointly maximize the importance and coherence of the summary.

2.1.4 Summarization Evaluation: A Harder Problem?

It is very important to know whether the constructed summaries meet the readers' information needs. Therefore, appropriate and reliable evaluation methods are very necessary to measure the performance of summarization systems. The goal of this section is to give an overview on the types of evaluation used to measure the performance of summarization systems.

Firstly, evaluating summary quality is a challenging task for several reasons. As Schriver [1989] pointed out, measuring text quality could be very subjective. For example, given the same summary about a news article, some readers may find it informative while some may not due to their background knowledge. Meanwhile, it is still unclear how to quantify many aspects of the summary quality, such as clarity, informativeness, or coherence. It is very common that different systems can generate comparable summaries with similar meanings by using disparate words, phrases, or sentences. More importantly, the ultimate goal of generating summaries is to improve users' reading experience and task performance, e.g. absorbing knowledge in a faster way. Therefore, in some scenarios, task-specific evaluation is required to measure how well the summaries serve the purpose.

In general, there are two types of evaluation for measuring the performance of summarization systems: **intrinsic evaluation** and **extrinsic evaluation**.

During the system development phase, quick evaluation and comparison are desired. Thus, **intrinsic evaluation** is designed for this goal. It is usually based on human judgments on the summary quality or comparison with human written gold-standard summaries. ROUGE [Lin and Hovy, 2003] is a widely used software for automatic summariztion evaluation on content coverage. It consists of a set of metrics calculating ngram overlap between a system generated summary and human written summaries, i.e. reference summaries. ROUGE scores have been used as the main evaluation measure for the summarization tasks in Document Understanding Conferences (DUC) and Text Analysis Conferences (TAC).

However, ROUGE is not a perfect metric. Especially, for lower-order ROUGE scores, they tend to detect significant differences among systems, though human judges find that they are comparable [Rankel et al., 2013]. Instead of measuring the ngram over-

lapping, Hovy et al. [2006] propose a Basic Elements (BEs) based evaluation method, where they represent each sentence as a set of semantic units, and calculate the coverage of BEs in the system summaries with regard to the reference summary. The BEs are defined as important syntactic structures, which can be achieved from constituent parse trees and dependency trees.

As mentioned above, human writers can come up with different summaries for the same input documents. Nevertheless, important facts tend to be mentioned more frequently across a set of human summaries for the same document(s) than secondary information. Based on the intuition that a better summary should cover more important facts, Nenkova and Passonneau [2004] develop the "pyramid" evaluation approach by using Summarization Content Units (SCUs) for summary content analysis. An SCU has a higher weight if it is mentioned more frequently by human summaries. Consequently, a summary covering SCUs with higher weights will have a higher pyramid score.

Intrinsic evaluation on other qualities of summarization systems, such as the linguistic quality, still very much relies on human judgment. For DUC or TAC conferences, human judges are asked to rate on various aspects of the system summaries, e.g. grammaticality, non-redundancy, clarity, or coherence.

Extrinsic evaluation usually carries out task-specific evaluations to measure whether the summarization systems enhance users' performance on specific tasks, such as absorbing complex knowledge, locating information, or making sense of massive amounts of data. The hypothesis is, better summarization systems should produce summaries that can help end-users more effectively complete the designed tasks. Nonetheless, extrinsic evaluation is time-consuming, and needs conscientious planning for the experiments. Previous work carries out extrinsic evaluation for the generated summaries on information retrieval tasks [Firmin Hand, 1997, Mani and Bloedorn, 1997, Jing et al.,

1998]. Time and accuracy are used to measure users' performance on relevance judgments of retrieved documents by reading their summaries. However, there are many factors that may affect users' performance, including summary length, summary presentation format, or query type. McKeown et al. [2005] design a task-based evaluation, where the users are asked to write a report for the documents with or without summaries displayed. Though they draw clear conclusions that summaries help with fact-gathering tasks, the influence from factors such as user interface, report length, or parameters in summarization system, is not well characterized.

Though recent summarization work continues to employ extrinsic evaluation [Christensen et al., 2014, Wang et al., 2015], there are still many questions left unanswered with the existing experimental designs. Follow-up studies are needed to provide deeper insights on which degree each factor affects the users' task performance, and more importantly, how to appropriately control those factors.

2.2 Genre- and Domain-Specific Summarization

In this section, we will introduce three types of summarization systems and their corresponding related work: focused meeting summarization (Section 2.2.1), user-generated content summarization (Section 2.2.2), and timeline generation (Section 2.2.3).

2.2.1 Focused Summarization for Spoken Meetings

Early research on spoken meeting summarization attempts to generate summaries for full dialogues [Xie et al., 2008, Garg et al., 2009, Riedhammer et al., 2010], which is called *generic* meeting summarization. Most work treats each meeting transcript as a

document, and applies existing extractive document summarization approaches, such as Maximal Marginal Relevance (MMR), Support Vector Machines (SVMs), Conditional Random Fields (CRFs) [Murray et al., 2005b, Xie et al., 2008, Galley, 2006], for meeting summarization. For unsupervised approaches, Garg et al. [2009] first cluster utterances according to their topics, then employ a graph-based algorithm to identify the important clusters and select representative sentences. Riedhammer et al. [2010] adopt the integer linear programming framework to select utterances that cover the most key phrases. Other related work on generic meeting summarization can be found in the book by Carenini et al. [2011].

Recently, the task of focused summarization has attracted more research attention. Supervised methods are investigated to identify key phrases or utterances for inclusion in the decision summary [Fernández et al., 2008, Bui et al., 2009]. Especially, Fernández et al. [2008] argue that phrases have the potential to yield higher recall and thus support better summaries. Input to their system, however, is narrowed down (manually) from the full set of decision-related dialogue acts to the subset that is useful for summarization. Based on this observation, we will describe a relation representation in Chapter 4, which can be output as a structured summary or used as content selection for abstractive summarization.

Our meeting summarization work lies in the area of generating abstractive summaries for conversations. As we have mentioned above, extractive approaches [Murray et al., 2005b, Xie et al., 2008, Galley, 2006] have been extensively investigated. Recent studies on summarizing consersational text have moved towards abstract-based approaches. Murray et al. [2010a] present an abstraction system consisting of interpretation and transformation steps. Utterances are mapped to a simple conversation ontology in the interpretation step according to their type, such as a decision or a problem. Then

an Integer Linear Programming approach is employed to select the utterances that cover more entities as determined by an external ontology. Their system makes the first effort towards abstractive summarization, however, it is still extractive in nature. Liu and Liu [2009] apply sentence compression on extracted summary utterances. Though some of the unnecessary words are dropped, the resulting compressions can still be ungrammatical and unstructured. According to their manual evaluation, even human compressions are barely satisfactory, and there is a noticeable gap between the human compressions and system compressions. Mehdad et al. [2014] propose an abstract generation framework based on word graphs, so that the system is able to compress utterances and merge information from different utterances simultaneously. However, this is still quite different from how humans write abstracts for different types of summaries. In the same spirit of Mehdad et al. [2014], Murray [2015] also build a word graph from a cluster of similar sentences. Then words are selected from the graph to generate the output sentences by using Markov Decision Processes.

2.2.2 Summarization for User-Generated Content

Traditionally, collecting public opinions is done by carrying out public opinion polls. As we have discussed, user-generated content, such as weblogs, comments, tweets, is a great resource to aggregate public's opinion and collect other users' thoughts on various topics. The availability of large amounts of user-generated content makes it possible to aggregate public opinion automatically with text analysis techniques. For example, O'Connor et al. [2010] find high correlation between some polling results and the sentiment word frequencies in the tweets.

In the spirit of facilitating common users' information seeking process, part of our

work is in line with understanding large amount of user-generated content and constructing opinion summaries of interest from it. Given that our goal is to digest opinions from different people, a concise and informative summary is desired.

There exists a large body of work on how to construct aspect-based opinion summaries, mainly from opinion mining community. An aspect-based summary is often constructed in a structured way. It consists of two main parts: target entities or aspects (or features) of the target, and the sentiment towards each entity or aspect. Hu and Liu [2004] first identify the frequent features for each product and then attach all the opinionated sentences to the corresponding feature. Lerman et al. [2009] use a similar framework, but they train a ranking SVM to better estimate the relevance of each sentence. Analogous to product feature extraction, Paul et al. [2010] exploit a topic model to discover contrastive viewpoints in phone surverys and editorials. Then a random walk based algorithm is designed to score sentence pairs for their contrastiveness. More related work can be found in the book by Liu [2012].

Structured summaries, that are organized with regard to the entities or aspects and the associated opinions, are appropriate for presenting opinions on specified products or topics. However, when the information being asked for is complex and hard to present in a structured way, a fluent text-based summary is more desired to show the relevant answers. This becomes one of the objectives in our work. Therefore, our work is more related to opinion summarization of user-generated content, such as blogs, user comments, or content from community Question Answering. The Text Analysis Conference (TAC) 2008 [Dang, 2008] first carried out an opinion summarization track on weblogs. Liu et al. [2008] manually construct taxonomies for questions in community QA. Summaries are generated by clustering sentences according to their polarity based on a small dictionary. Tomasoni and Huang [2010] introduce coverage and quality constraints on

the sentences, and utilize an Integer Linear Programming framework to select sentences.

There is a growing interest in generating article summaries informed by social context. Existing work focuses on learning users' interests from comments and incorporates the learned information into a news article summarization system [Hu et al., 2008]. For example, Hu et al. [2008] estimate word importance with comments, and compute the average word importance of each sentence for use in the ranking function. The article summary is then constructed by selecting top ranked sentences progressively. Zhao et al. [2013] instead estimate word distributions from tweets, and bias a Page Rank algorithm to give higher restart probability to sentences with similar distributions. Generating tweet+article summaries has been recently investigated in Yang et al. [2011]. They propose a factor graph to allow sentences and tweets to mutually reinforce each other. Gao et al. [2012] exploit a co-ranking model to identify sentence-tweet pairs with complementary information estimated from a topic model.

2.2.3 Timeline Generation

One crucial reason for constructing a summary is to help users absorb information in an efficient way. This dissertation further improves users' reading experience by extending traditional multi-document summarization with temporal relations on events. Temporal information has long been considered as helpful for improving summary quality by displaying the connected events in chronological order or improving the estimation on information relevancy or recency [Goldstein et al., 2000, Allan et al., 2001, Demartini et al., 2010, Ng et al., 2014].

Timeline generation techniques have been proposed to display a series of relevant event summaries in chronological order. Previous work has been focusing on modeling

the important properties of timelines. For instance, the system by Chieu and Lee [2004] ranks sentences according to "burstiness" and "interestingness" estimated by a likelihood ratio test. Yan et al. [2011] explore an optimization framework that maximizes the relevance, coverage, diversity, and coherence of the timeline. Neither system has leveraged the social context. Built on the same system, Zhao et al. [2013] utilized a Page Rank based algorithm to upweight the frequently mentioned content in tweets.

Our event threading algorithm is also inspired by work on topic detection and tracking (TDT) [Allan et al., 1998], where efforts are made for document-level link detection and topic tracking. Similarly, Nallapati et al. [2004] investigate event threading for articles, where they predict linkage based on causal and temporal dependencies. Their work demonstrates the utility of event-focused organization in timeline analysis, although they do not proceed to generate summaries. Shahaf et al. [2012] instead seek to connect articles into one coherent graph. To the best of our knowledge, we are the first to study sentence-level event threading. Gillenwater et al. [2012] present techniques for selecting and ordering news articles to describe causal paths from one news report to another. They identify important metrics for maintaining coherent structure, such as the contribution of individual words to the lexical similarity between two consecutive articles.

2.3 Sentiment Analysis for User-Generated Content

This section first briefly introduces the popular research area of sentiment analysis (Section 2.3.1), which is followed by related work for agreement and disagreement detection in online discussions (Section 2.3.2).

2.3.1 Why Do We Care About Sentiment?

As we have described in the introduction of this dissertation, user-generated content on the Internet has become a prevailing source for various information seeking purposes. For instance, users can find out other people's opinion on various topics by reading all the relevant information. Given the massive amount of text available, it would be difficult to read all of the opinions and determine the relevant ones. Automatic systems are developed to address this issue. Early work on sentiment analysis has focused on identifying the valence of a piece of text, i.e. whether the text contains positive, negative, or neutral sentiment. Experiments have been conducted on movie reviews [Pang et al., 2002] or product reviews [Dave et al., 2003, Hu and Liu, 2004]. To have a detailed understanding of users' opinion on a topic of interest, opinion mining techniques are proposed to produce a summary consisting of concrete aspects that people express opinions on [Choi et al., 2005, Stoyanov and Cardie, 2006, Breck et al., 2007]. More related work on sentiment analysis and opinion mining for various types of text can be found in the book by Pang and Lee [2008].

2.3.2 Agreement and Disagreement Detection in Online Social Interactions

In this dissertation, we will focus on understanding the sentiment or opinion expressed in online conversations. This is important because it has become prevalent for Internet users to use online discussion forums to express their opinions and argue with others on critical social or political issues.

Previous work has shown that sentiment analysis can be used as a key enabling

technique in a number of conversation-based applications. Some early work studies the attitudes in spoken meetings [Galley et al., 2004, Hahn et al., 2006] or broadcast conversations [Wang et al., 2011] by using Conditional Random Fields (CRF) [Lafferty et al., 2001a]. Galley et al. [2004] employ Conditional Markov models to detect if discussants reach an agreement in spoken meetings. Each state in their model is an individual turn and prediction is made on the turn-level. In the same spirit, Wang et al. [2011] also propose a sequential model based on CRF for detecting agreements and disagreements in broadcast conversations, where they primarily show the efficiency of prosodic features. While we also exploit a sequential model extended from CRFs, our predictions are made for each sentence or segment rather than at the turn-level. Moreover, we experiment with online discussion datasets that exhibit a more realistic distribution of disagreement vs. agreement, where much more disagreement is observed due to its function and the relation between the participants. This renders the detection problem more challenging.

Only recently, agreement and disagreement detection is studied for online discussion, especially for online debate. For example, Misra and Walker [2013] study the effectiveness of topic-independent features, e.g. discourse cues indicating agreement or negative opinion. Those cues, which serve a similar purpose as a sentiment lexicon, are also constructed manually. In our work, we create an online discussion lexicon automatically and construct sentiment features based on the lexicon. Also targeting online debate, Yin et al. [2012] train a logistic regression classifier with features aggregating posts from the same participant to predict the sentiment for each individual post. This approach works only when the speaker has enough posts on each topic, which is not applicable to newcomers. Hassan et al. [2010] focus on predicting the attitude of participants towards each other. They relate the sentiment words to the second person pronoun, which produces strong baselines. We also adopt their baselines in our work. Although

there are available datasets with (dis)agreement annotated on Wikipedia talk pages, we are not aware of any published work that utilizes these annotations. Dialogue act recognition on talk pages [Ferschke et al., 2012] might be the most related. To improve the performance on disagreement detection in online discussions, Allen et al. [2014] also study features derived from rhetorical structures, and find that discourse relations are very indicative for this task.

While detecting agreement and disagreement in conversations is useful on its own, it is also a key component for related tasks, such as stance prediction [Thomas et al., 2006, Somasundaran and Wiebe, 2009, Walker et al., 2012b] and subgroup detection [Hassan et al., 2012, Abu-Jbara et al., 2012]. For instance, Thomas et al. [2006] train an agreement detection classifier with Support Vector Machines on congressional floor-debate transcripts to determine whether the speeches represent support of, or opposition to, the proposed legislation. Somasundaran and Wiebe [2009] design various sentiment constraints for inclusion in an Integer Linear Programming framework for stance classification. For subgroup detection, Abu-Jbara et al. [2012] use the polarity of the expressions in the discussions and partition discussants into subgroups based on the intuition that people in the same group should mostly agree with each other. Though those works highly rely on the component of agreement and disagreement detection, the evaluation is always performed on the final application only.

CHAPTER 3

MEETING SUMMARIZATION: BEYOND UTTERANCE EXTRACTION

In this chapter, we will present two focused meeting summarization frameworks beyond utterance extraction. The system summaries are represented as key words or relations, which can be used as the basis of abstractive summary generation. This chapter corresponds to our contribution on summarization for text with inherent noise.

3.1 Introduction

For better or worse, meetings play an integral role in most of our daily lives — they let us share information and collaborate with others to solve a problem, to generate ideas, and to weigh options. Not surprisingly then, there is growing interest in developing automatic methods for meeting summarization (e.g., Zechner [2002], Maskey and Hirschberg [2005], Galley [2006], Lin and Chen [2010], Murray et al. [2010a]). We tackle the task of *focused meeting summarization*, i.e., generating summaries of a particular aspect of a meeting rather than of the meeting as a whole [Carenini et al., 2011]. For example, one might want a summary of just the DECISIONS made during the meeting, the ACTION ITEMS that emerged, the IDEAS discussed, or the HYPOTHESES put forth, etc.

Meeting conversation is intrinsically different from well-written text, as meetings may not be well organized and most utterances have low density of salient content. Therefore, multiple problems need to be addressed for speech summarization. Consider the sample dialogue snippet in Figure 3.1 from the AMI meeting corpus [Carletta et al., 2005]. Only *decision-related dialogue acts (DRDAs)* — utterances at least one decision

made in the meeting[1] — are listed and ordered by time. Each DRDA is labeled numerically according to the decision it supports; so the second and third utterances (in **bold**) support DECISION 2, as do the fifth utterance in the snippet. Manually constructed *decision abstracts* for each decision are shown at the bottom of the figure. [2] These constitute the *decision-focused summary* for the snippet.

Besides the prevalent dialogue phenomena (such as "Uh I'm kinda liking" in Figure 3.1), disfluencies and off-topic expressions, we notice that single utterance is usually not informative enough to form a decision. For instance, no single DRDA associated with DECISION 4 corresponds all that well with its decision abstract: "pushbuttons", "menu button" and "Pre-set channels" are mentioned in separate DAs. As a result, extractive summarization methods that select individual utterance to form the summary will perform poorly.

Furthermore, it is difficult to identify the core topic when multiple topics are discussed in one utterance. For example, all of the bold DRDAs supporting DECISION 2 contain the word "latex". However, the last DA in bold also mentions "bigger impact" and "the scroll wheel", which are not specifically relevant for DECISION 2. Though this problem can be approached by training a classifier to identify the relevant phrases and ignore the irrelevant ones or dialogue phenomena, it needs expensive human annotation and is limited to the specific domain.

In this chapter, we will study two summarization frameworks that generate focused meeting summaries beyond utterance extraction. In Section 3.2, we first study the unsu-

[1]These DRDAs are annotated in the AMI corpus and usually contain the decision content. They are similar, but not completely equivalent, to the *decision dialogue acts (DDAs)* of Bui et al. [2009], Fernández et al. [2008], Frampton et al. [2009].

[2]Murray et al. [2010b] show that users much prefer *abstractive summaries* over extracts when the text to be summarized is a conversation. In particular, extractive summaries drawn from group conversations can be confusing to the reader without additional context; and the noisy, error-prone, disfluent text of speech transcripts is likely to result in extractive summaries with low readability.

A:We decided our target group is the focus on who can afford it , (1)
B:Uh I'm kinda liking the idea of latex , if if spongy is the in thing . (2)
B:what I've seen , just not related to this , but of latex cases before , is that [vocalsound] there's uh like a hard plastic inside , and it's just covered with the latex . (2)
C:Um [disfmarker] And I think if we wanna keep our costs down , we should just go for pushbuttons , (3)
D:but if it's gonna be in a latex type thing and that's gonna look cool , then that's probably gonna have a bigger impact than the scroll wheel . (2)
A:we're gonna go with um type pushbuttons , (3)
A:So we're gonna have like a menu button , (4)
C:uh volume , favourite channels , uh and menu . (4)
A:Pre-set channels (4)

Decision Abstracts (Summary)
DECISION 1: The target group comprises of individuals
who can afford the product.
DECISION 2: The remote will have a latex case.
DECISION 3: The remote will have pushbuttons.
DECISION 4: The remote will have a power button, volume
buttons, channel preset buttons, and a menu button.

Figure 3.1: A clip of a meeting from the AMI meeting corpus [Carletta et al., 2005]. A, B, C and D refer to distinct speakers; the numbers in parentheses indicate the associated meeting decision: DECISION 1, 2, 3 or 4. Also shown is the gold-standard (manual) abstract (summary) for each decision.

pervised topic modeling-based approaches for decision summarization in meetings by identifying a concise set of key words or phrases, which can either be output as a compact summary or be a starting point to generate abstractive summaries. Specifically, as a step towards creating the abstractive summaries, we propose a token-level rather than sentence-level framework for identifying components of the summary. Experimental results show that, compared to the sentence ranking based summarization algorithms, our token-level summarization framework can better identify the summary-worthy words and remove the redundancies.

Moreover, rather than employing supervised learning methods that rely on costly manual annotation, we explore and evaluate topic modeling approaches of different

granularities for the unsupervised decision summarization at both the token-level and dialogue act-level. We investigate three topic models — Local LDA (LocalLDA) [Brody and Elhadad, 2010], Multi-grain LDA (MG-LDA) [Titov and McDonald, 2008] and Segmented Topic Model (STM) [Du et al., 2010] — which can utilize the latent topic structure on utterance level instead of document level. Under our proposed token-level summarization framework, three fine-grained models outperform the basic LDA model and two extractive baselines that select the longest and the most representative utterance for each decision, respectively. (ROUGE-SU4 F score of 14.82% for STM vs. 13.58% and 13.46% for the baselines, given the perfect clusterings of DRDAs.)

We also investigate the role of context in our token-level summarization framework. For the given clusters of DRDAs, We study two types of context information — the DAs preceding and succeeding a DRDA and DAs of high TF-IDF similarity with a DRDA. We also investigate two ways to select relevant words from the context DA. Experimental results show that two types of context have comparable effect, but selecting words from the dominant topic of the center DRDA performs better than from the dominant topic of the context DA. Moreover, by leveraging context, the recall exceeds the provided upperbound's recall (ROUGE-1 recall: 48.10% vs. 45.05% for upperbound by using DRDA only) although the F scores decrease after adding context information. Finally, we show that when the true DRDA clusterings are not available, adding context can improve both the recall and F score.

In Section 3.3, we will present another unsupervised framework for focused meeting summarization that supports the generation of abstractive summaries. We view the problem as an **information extraction** task and hypothesize that existing methods for domain-specific relation extraction can be modified to identify salient phrases for use in generating abstractive summaries.

C: Say the standby button is quite kinda separate from all the other functions. (1)
C: Maybe that could *be* [a little apple]. (1)
C: It seems like you're gonna *have* [rubber cases], as well as [buttons]. (2)
A: [Rubber buttons] *require* [rubber case]. (2)
A: You could *have* [your company badge] and [logo]. (3)
A: I mean a lot of um computers for instance like like on the one you've got there, it actually *has* a sort of um [stick on badge]. (3)
C: Shall we *go* [for single curve], just to compromise? (2)
B: We'll *go* [for single curve], yeah. (2)
C: And the rubber push buttons, rubber case. (2)
D: And then are we going for sort of [one button] *shaped* [like a fruit]. <vocalsound> Or veg. (1)
D: Could *be* [a red apple], yeah. (1)

Decision Abstracts (Summary)
DECISION 1: The group decided to make the standby button in the shape of an apple.
DECISION 2: The remote will also feature a rubber case and rubber buttons, and a single-curved design.
DECISION 3: The remote will feature the company logo, possibly in a sticker form.

Figure 3.2: Clip from the AMI meeting corpus [Carletta et al., 2005]. A, B, C and D refer to distinct speakers; the numbers in parentheses indicate the associated meeting decision: DECISION 1, 2 or 3. Also shown is the gold-standard (manual) abstract (summary) for each decision. Colors indicate overlapping vocabulary between utterances and the summary. Underlining, *italics*, and [bracketing] are decscribed in the running text.

Very generally, information extraction methods identify a lexical "trigger" or "indicator" that evokes a relation of interest and then employ syntactic information, often in conjunction with semantic constraints, to find the "target phrase" or "argument constituent" to be extracted. Relation instances, then, are represented by **indicator-argument** pairs [Chen et al., 2011].

Consider another example in Figure 3.2. Notice that many portions of the DRDAs are not relevant to the decision itself: they often begin with phrases that identify the utterance within the discourse as potentially introducing a decision (e.g., "Maybe that could be", "It seems like you're gonna have"), but do not themselves describe the deci-

sion. We will refer to this portion of a DRDA (underlined in Figure 3.2) as the **Decision Cue**. Moreover, the decision cue is generally directly followed by the actual **Decision Content** (e.g., "be a little apple", "have rubber cases"). Decision Content phrases are denoted in Figure 3.2 via italics and square brackets. Importantly, it is just the decision content portion of the utterance that should be considered for incorporation into the focused summary.

Some possible indicator-argument pairs for identifying the Decision Content phrases are displayed in the dialogue sample in Figure 3.2. Content **indicator** words are shown in *italics*; the Decision Content target phrases are the **arguments**. For example, in the fourth DRDA, "require" is the indicator, and "rubber buttons" and "rubber case" are both arguments. Although not shown in Figure 3.2, it is also possible to identify relations that correspond to the **Decision Cue** phrases. Consider, for example, the phrases underlined in the sixth and seventh DRDAs. "I mean" and "shall we" are two typical Decision Cue phrases where "mean" and "shall" are possible indicators with "I" and "we" as their arguments, respectively.

Specifically, we still focus on the task of *decision summarization* and, as in previous work in meeting summarization (e.g., Fernández et al. [2008], Wang and Cardie [2011]), assume that all decision-related utterances (DRDAs) have been identified. We adapt the unsupervised relation learning approach of Chen et al. [2011] to separately identify relations associated with decision cues vs. the decision content within DRDAs by defining a new set of task-specific constraints and features to take the place of the domain-specific constraints and features of the original model. Output of the system is a set of extracted indicator-argument decision content relations (see the "OUR METHOD" sample summary of Table 3.9) that can be used as the basis of the decision abstract.

We evaluate the approach (using the AMI corpus [Carletta et al., 2005]) under two

input settings — in the **True Clusterings** setting, we assume that the DRDAs for each meeting have been perfectly grouped according to the decision(s) each supports; in the **System Clusterings** setting, an automated system performs the DRDA-decision pairing. The results show that the relation-based summarization approach outperforms two extractive summarization baselines that select the longest and the most representative utterance for each decision, respectively. (ROUGE-1 F score of 37.47% vs. 32.61% and 33.32% for the baselines given the True Clusterings of DRDAs.) Moreover, our approach performs admirably in comparison to two supervised learning alternatives (scores of 35.61% and 40.87%) that aim to identify the important **tokens** to include in the decision abstract given the DRDA clusterings. In contrast to our approach which is transferable to different domains or tasks, these methods would require labeled data for retraining for each new meeting corpus.

Finally, in order to compare our approach to another *relation-based* summarization technique, we modify the multi-document summarization system of Hachey [2009] to the single-document meeting scenario. Here again, our proposed approach performs better (37.47% vs. 34.69%). Experiments under the System Clusterings setting produce the same overall results, albeit with lower scores for all of the systems and baselines.

3.2 Token-Level Representation via Unsupervised Topic Modeling

In this section, we describe a token-level decision summarization method based on unsupervised topic modeling approaches (Sections 3.2.1 and 3.2.2). Experimental setup and results are presented in Sections 3.2.3 and 3.2.4.

3.2.1 Summarization Frameworks

We first present our proposed token-level decision summarization framework — **DomSum** — which utilizes latent topic structure in utterances to extract words from **Dom**inant Topic to form **Sum**maries. In Section 3.2.2, we describe four existing sentence scoring metrics denoted as *OneTopic, MultiTopic, TMMSum* and *KLSum* which are also based on latent topic distributions. We adopt them to the utterance-level summarization for comparison in Section 3.2.4.

Token-level Summarization Framework

Domsum takes as input the clusters of DRDAs (with or without additional context DAs), the topic distribution for each DA and the word distribution for each topic. The output is a set of topic-coherent summary-worthy words which can be used directly as the summary or to further generate abstractive summary. We introduce DomSum in two steps according to its input: taking clusters of DRDAs as the input and with additional context information.

DRDAs Only. Given clusters of DRDAs, we use Algorithm 1 to produce the token-level summary for each cluster. Generally, Algorithm 1 chooses the topic with the highest probability as the *dominant topic* given the dialogue act (DA). Then it collects the words with a high joint probability with the dominant topic from that DA.

Leveraging Context. For each DRDA (denoted as "*center DA*"), we study two types of context information (denoted as "*context DAs*"). One is adjacent DAs, i.e., immediately preceding and succeeding DAs, the other is the DAs having top TF-IDF similarities

```
Input  : Cluster C = {DA_i}, P(T_j|DA_i), P(w_k|T_j)
Output: Summary

Summary← Φ (empty set)
foreach DA_i in C do
    DomTopic← max_{T_j} P(T_j|DA_i) (*)
    Candidate← Φ
    foreach word w_k in DA_i do
        SampleTopic← max_{T_j} P(w_k|T_j)P(T_j|DA_i)
        if DomTopic == SampleTopic then
            Candidate ← Union(Candidate, w_k)
        end
    end
    Summary ← Union(Summary, Candidate)
end
```

Algorithm 1: DomSum – The token-level summarization framework. DomSum takes as input the clusters of DRDAs and related probability distributions.

with the center DA. Context DAs are added into the cluster the corresponding center DA in.

We also study two criteria of word selection from the context DAs. For each context DA, we can take the words appearing in the dominant topic of either this context DA or its center DRDA. We will show in Section 6.1 that the latter performs better as it produces more topic-coherent summaries. Algorithm 1 can be easily modified to leverage context DAs by updating the input clusters and assigning the proper dominant topic for each DA accordingly — this changes the step $(*)$ in Algorithm 1.

Utterance-level Summarization Metrics

We also adopt four sentence scoring metrics based on the latent topic structure for extractive summarization. Though they are developed on different topic models, given the desired topic distributions as input, they can rank the utterances according to their importance and provide utterance-level summaries for comparison.

OneTopic and MultiTopic. In Bhandari et al. [2008], several sentence scoring functions are introduced based on Probabilistic Latent Semantic Indexing. We adopt two metrics, which are *OneTopic* and *MultiTopic*. For OneTopic, topic T with highest probability $P(T)$ is picked as the central topic per cluster C. The score for DA in C is:

$$P(DA|T) = \frac{\sum_{w \in DA} P(T|DA, w)}{\sum_{DA' \in C, w \in DA'} P(T|DA', w)},$$

MultiTopic modifies OneTopic by taking all of the topics into consideration. Given a cluster C, DA in C is scored as:

$$\sum_{T} P(DA|T)P(T) = \sum_{T} \frac{\sum_{w \in DA} P(T|DA, w)}{\sum_{DA' \in C, w \in DA'} P(T|DA', w)} P(T)$$

TMMSum. Chen and Chen [2008] propose a Topical Mixture Model (TMM) for speech summarization, where each dialogue act is modeled as a TMM for generating the document. TMM is shown to provide better utterance-level extractive summaries for spoken documents than other conventional unsupervised approaches, such as Vector Space Model (VSM) [Gong and Liu, 2001], Latent Semantic Analysis (LSA) [Gong and Liu, 2001] and Maximum Marginal Relevance (MMR) [Murray et al., 2005a]. The importance of a sentence S can be measured by its generative probability $P(D|S)$, where D is the document S belongs to. In our experiments, one decision is made per cluster of DAs. So we adopt their scoring metric to compute the generative probability of the cluster C for each DA:

$$P(C|DA) = \prod_{w_i \in C} \sum_{T_j} P(w_i|T_j)P(T_j|DA),$$

KLSum. Kullback-Lieber (KL) divergence is explored for summarization in Haghighi and Vanderwende [2009] and Lin et al. [2010], where it is used to measure the distance of distributions between the document and the summary. For a cluster C of DAs, given a length limit θ, a set of DAs S is selected as:

$$S^* = \arg\min_{S:|S|<\theta} KL(P_C||P_S) = \arg\min_{S:|S|<\theta} \sum_{T_i} P(T_i|C) log \frac{P(T_i|C)}{P(T_i|S)}$$

3.2.2 Topic Models

In this section, we briefly describe the three fine-grained topic models employed to compute the latent topic distributions on utterance level in the meetings. According to the input of Algorithm 1, we are interested in estimating the topic distribution for each DA $P(T|DA)$ and the word distribution for each topic $P(w|T)$. For MG-LDA, $P(T|DA)$ is computed as the expectation of local topic distributions with respect to the window distribution.

Local LDA

Local LDA (LocalLDA) [Brody and Elhadad, 2010] uses almost the same probabilistic generative model as Latent Dirichlet Allocation (LDA) [Blei et al., 2003], except that it treats each sentence as a separate document[3]. Each DA d is generated as follows:

- For each topic k:
  - Choose word distribution: $\phi_k \sim Dir(\beta)$

[3]For the generative process of LDA, the DAs in the same meeting make up the document, so "each DA" is changed to "each meeting" in LocalLDA's generative process.

- For each DA d:
  - Choose topic distribution: $\theta_d \sim Dir(\alpha)$
  - For each word w in DA d:
    - Choose topic: $z_{d,w} \sim \theta_d$
    - choose word: $w \sim \phi_{z_{d,w}}$

Multi-grain LDA

Multi-grain LDA (MG-LDA) [Titov and McDonald, 2008] can model both the meeting specific topics (e.g. the design of a remote control) and various concrete aspects (e.g. the cost or the functionality). The generative process is:

- Choose a global topic distribution: $\theta_m^{gl} \sim Dir(\alpha^{gl})$
- For each sliding window v of size T:
  - Choose local topic distribution: $\theta_{m,v}^{loc} \sim Dir(\alpha^{loc})$
  - Choose granularity mixture: $\pi_{m,v} \sim Beta(\alpha^{mix})$
- For each DA d:
  - choose window distribution: $\psi_{m,d} \sim Dir(\gamma)$
- For each word w in DA d of meeting m:
  - Choose sliding window: $v_{m,w} \sim \psi_{m,d}$
  - Choose granularity: $r_{m,w} \sim \pi_{m,v_{m,w}}$
  - If $r_{m,w} = gl$, choose global topic: $z_{m,w} \sim \theta_m^{gl}$
  - If $r_{m,w} = loc$, choose local topic: $z_{m,w} \sim \theta_{m,v_{m,w}}^{loc}$
  - Choose word w from the word distribution: $\phi_{z_{m,w}}^{r_{m,w}}$

Segmented Topic Model

The last model we utilize is Segmented Topic Model (STM) [Du et al., 2010], which jointly models document- and sentence-level latent topics using a two-parameter Poisson Dirichlet Process (PDP). Given parameters α, γ, Φ and PDP parameters a, b, the generative process is:

- Choose distribution of topics: $\theta_m \sim Dir(\alpha)$
- For each dialogue act d:
  - Choose distribution of topics: $\theta_d \sim PDP(\theta_m, a, b)$
- For each word w in dialogue act d:
  - Choose topic: $z_{m,w} \sim \theta_d$
  - Choose word: $w \sim \phi_{z_{m,w}}$

3.2.3 Experimental Setup

The Corpus. We evaluate our approach on the AMI meeting corpus [Carletta et al., 2005] that consists of 140 multi-party meetings. The 129 scenario-driven meetings involve four participants playing different roles on a design team. A short (usually one-sentence) abstract is manually constructed to summarize each decision discussed in the meeting and used as gold-standard summaries in our experiments.

System Inputs. Our summarization system requires as input a partitioning of the DRDAs according to the decision(s) that each supports (i.e., one cluster of DRDAs per

decision). As mentioned earlier, we assume for all experiments that the DRDAs for each meeting have been identified. For evaluation we consider two system input settings. In the **True Clusterings** setting, we use the AMI annotations to create perfect partitionings of the DRDAs as the input; in the **System Clusterings** setting, we employ a hierarchical agglomerative clustering algorithm used for this task in our work [Wang and Cardie, 2011]. The clustering method groups DRDAs according to their LDA topic distribution similarity. As better approaches for DRDA clustering become available, they could be employed instead.

Evaluation Metric. To evaluate the performance of various summarization approaches, we use the widely accepted ROUGE [Lin and Hovy, 2003] metrics. We use the stemming option of the ROUGE software at `http://berouge.com/` and remove stopwords from both the system and gold-standard summaries, same as Riedhammer et al. [2010] do.

Inference and Hyperparameters We use the implementation from Lu et al. [2011] for the three topic models in Section 3.2.2. The collapsed Gibbs Sampling approach [Griffiths and Steyvers, 2004] is exploited for inference. Hyperparameters are chosen according to Brody and Elhadad [2010], Titov and McDonald [2008] and Du et al. [2010]. In LDA and LocalLDA, α and β are both set to 0.1 . For MG-LDA, α^{gl}, α^{loc} and α^{mix} are set to 0.1; γ is 0.1 and the window size T is 3. And the number of local topic is set as the same number of global topic as discussed in Titov and McDonald [2008]. In STM, α, a and b are set to 0.5, 0.1 and 1, respectively.

Baselines and Comparisons

We compare our token-level summarization framework based on the fine-grained topic models to (1) two unsupervised baselines, (2) token-level summarization by LDA, (3) utterance-level summarization by Topical Mixture Model (TMM) [Chen and Chen, 2008], (4) utterance-level summarization based on the fine-grained topic models using existing metrics (Section 3.2.1), (5) two supervised methods, and (6) an upperbound derived from the AMI gold standard decision abstracts. (1) and (6) are described below, others will be discussed in Section 3.2.4.

The LONGEST DA **Baseline.** As in Riedhammer et al. [2010], this baseline simply selects the longest DRDA in each cluster as the summary. Thus, it performs utterance-level decision summarization. This baseline and the next allow us to determine summary quality when summaries are restricted to a single utterance.

The PROTOTYPE DA **Baseline.** The second baseline selects the decision cluster prototype (i.e., the DRDA with the largest TF-IDF similarity with the cluster centroid) as the summary.

Upperbound. We also compute an upperbound that reflects the gap between the best possible extractive summaries and the human-written abstracts according to the ROUGE score: for each cluster of DRDAs, we select the words that also appear in the associated decision abstract.

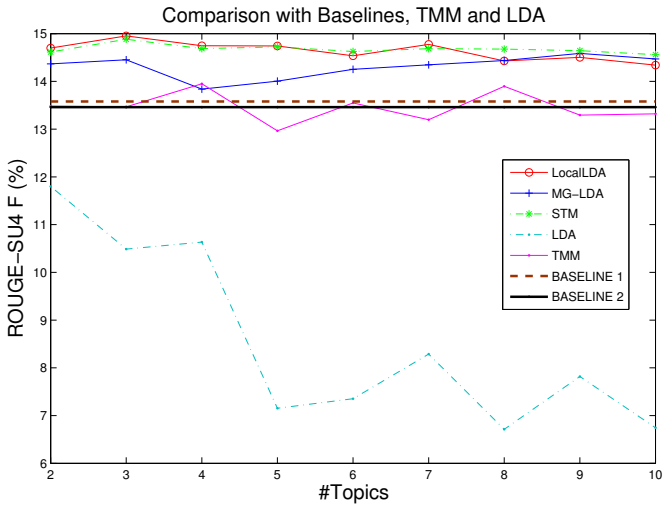


Figure 3.3: With true clusterings of DRDAs as the input, we use DomSum to compare the performance of LocalLDA, MGLDA and STM against two baselines, LDA and TMM. "# topic" indicates the number of topics for the model. For MGLDA, "# topic" is the number of local topics.

3.2.4 Results

True Clusterings

How do fine-grained topic models compare to basic topic models or baselines? Figure 3.3 demonstrates that by using the DomSum token-level summarization framework, the three fine-grained topic models uniformly outperform the two non-trivial baselines and TMM [Chen and Chen, 2008] (reimplemented by us) that generates utterance-level summaries. Moreover, the fine-grained models also beat basic LDA under the same DomSum token-level summarization framework. This shows the fine-grained topic models that discover topic structures on utterance-level better identify gist information.

Can the proposed token-level summarization framework better identify important words and remove redundancies than utterance selection methods? Figure 3.4

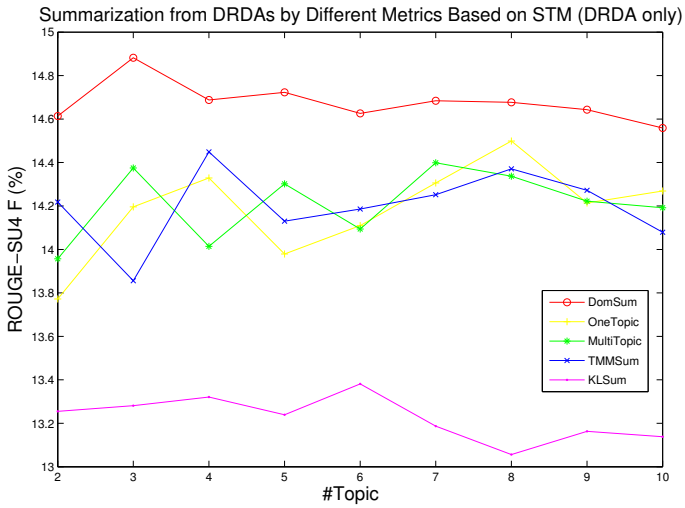


Figure 3.4: With true clusterings of DRDAs as the input, DomSum is compared with four DA-level summarization metrics using topic distributions from STM. Results from LocalLDA and MGLDA are similar so they are not displayed.

demonstrates the comparison results for our DomSum token-level summarization framework with four existing utterance scoring metrics discussed in Section 3.2.1, namely OneTopic, MultiTopic, TMMSum and KLSum. The utterance with highest score is extracted to form the summary. LocalLDA and STM are utilized to compute the input distributions, i.e., $P(T|DA)$ and $P(w|T)$. From Figure 3.4, DomSum yields the best F scores which shows that the token-level summarization approach is more effective than utterance-level methods.

Which way is better for leveraging context information? We explore two types of context information. For adjacent content (*Adj* in Figure 3.5), 5 DAs immediately preceding and 5 DAs succeeding the center DRDA are selected. For TF-IDF context (*TFIDF* in Figure 3.5), 10 DAs of highest TF-IDF similarity with the center DRDA are taken. We also explore two ways to extract summary-worthy words from the context DA — selecting words from the dominant topic of either the center DA (denoted as "One" in parentheses in Figure 3.5) or the current context DA (denoted as "multi" in parentheses

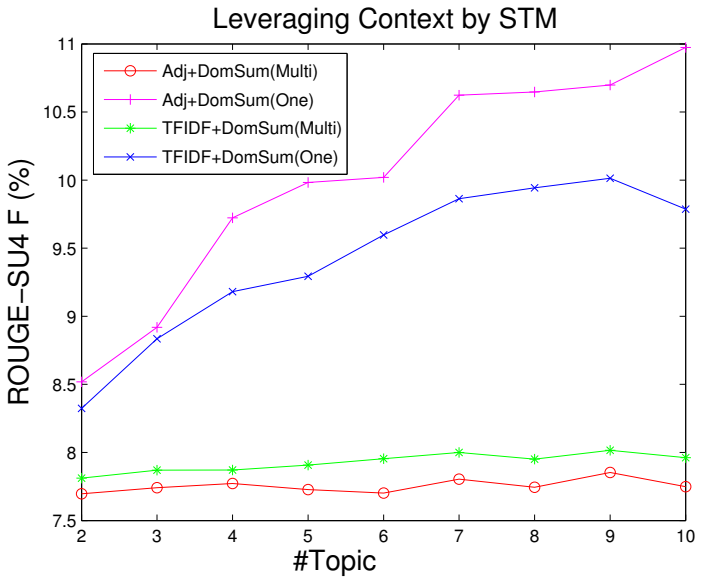


Figure 3.5: Under DomSum framework, two types of context information are added: Adjacent DA ("Adj") and DAs with high TFIDF similarities ("TFIDF"). For each context DA, selecting words from the dominant topic of center DA ("One") or the current context DA ("Multi") are investigated.

in Figure 3.5).

Figure 3.5 indicates that the two types of context information do not have significant difference, while selecting the words from the dominant topic of the center DA results in better ROUGE-SU4 F scores. Notice that compared with Figure 3.4, the results in Figure 3.5 have lower F scores when using the true clusterings of DRDAs. This is because context DAs bring in relevant words as well as noisy information. We will show that when true clusterings are not available, the context information can boost both recall and F score.

How does the token-level summarization framework compare to utterance selection methods for leveraging context? We also compare the ability of leveraging context of DomSum to utterance scoring metrics, i.e., OneTopic and MultiTopic. 5 DAs preceding and 5 DAs succeeding the center DA are added as context information. For

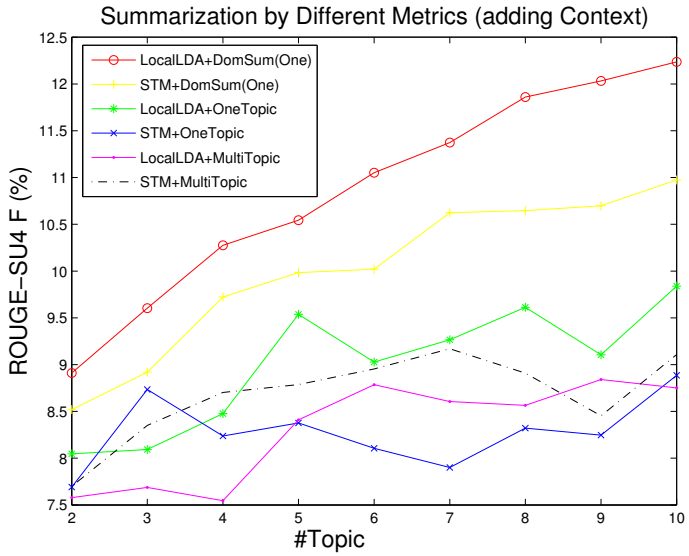


Figure 3.6: By using adjacent DAs as context, DomSum is compared with two DA-level summarization metrics: OneTopic and MultiTopic. For DomSum, the words of context DA from dominant topic of the center DA ("One") is selected; For OneTopic and MultiTopic, three top ranked DAs are selected.

context DA under DomSum, we select words from the dominant topic of the center DA (denoted as "One" in parentheses in Figure 3.6). For OneTopic and MultiTopic, the top 3 DAs are extracted as the summary. Figure 3.6 demonstrates the combination of LocalLDA and STM with each of the metrics. DomSum, as a token-level summarization metrics, dominates other two metrics in leveraging context.

How does our approach perform compare with supervised learning approaches? For a better comparison, we also provide summarization results by using supervised systems along with an upperbound. We use Support Vector Machines [Joachims, 1998] with RBF kernel and order-1 Conditional Random Fields [Lafferty et al., 2001b] — trained with the same features as our previous work [Wang and Cardie, 2011] to identify the summary-worthy **tokens** to include in the abstract. A three-fold cross validation is conducted for both methods. ROUGE-1, ROUGE-2 and ROUGE-SU4 scores are listed in Table 3.1.

	True Clusterings				
	R-1			R-2	R-SU4
	PREC	REC	F1	F1	F1
Baselines					
Longest DA	34.06	31.28	32.61	12.03	13.58
Prototype DA	40.72	28.21	33.32	12.18	13.46
Supervised Methods					
CRF	52.89	26.77	35.53	11.48	14.03
SVM	43.24	37.92	40.39	12.78	16.24
Our Approach					
5 topics					
LocalLDA	35.18	38.92	36.95	12.33	14.74
+ *context*	17.26	45.34	25.00	8.40	11.05
STM	34.06	41.30	**37.32**	**12.42**	**14.82**
+ *context*	15.60	***48.10***	23.56	8.16	9.98
10 topics					
LocalLDA	36.20	36.81	36.50	12.04	14.34
+ *context*	21.82	41.57	28.62	9.61	12.24
STM	34.15	40.83	37.19	12.40	14.56
+ *context*	17.87	46.57	25.82	8.89	10.97
Upperbound	100.00	***45.05***	62.12	33.27	34.89

Table 3.1: ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) scores for our proposed token-level summarization approaches along with two baselines, supervised methods and the Upperbound (only using DRDAs). — all use True Clusterings

From Table 3.1, our token-level summarization approaches based on LocalLDA and STM are shown to outperform the baselines and even the CRF. Meanwhile, by adding context information, both LocalLDA and STM can get better ROUGE-1 recall than the supervised methods, even higher than the provided upperbound which is computed by only using DRDAs. This shows that the DomSum framework can leverage context to compensate the summaries.

System Clusterings

Results using the **System Clusterings** (Table 3.2) present similar findings, though all of the system and baseline scores are lower. By adding context information, the token-level summarization approaches based on fine-grained topic models compare favorably to the supervised methods in F scores, and also get the best ROUGE-1 recalls.

	System Clusterings				
	R-1			**R-2**	**R-SU4**
	PREC	REC	F1	F1	F1
Baselines					
Longest DA	17.06	11.64	13.84	2.76	3.34
Prototype DA	18.14	10.11	12.98	2.84	3.09
Supervised Methods					
CRF	46.97	15.25	23.02	6.09	9.11
SVM	39.05	18.45	25.06	6.11	9.82
Our Approach					
5 topics					
LocalLDA	25.57	16.57	20.11	4.03	5.87
+ *context*	20.68	25.96	23.02	3.09	4.48
STM	24.15	17.82	20.51	4.03	5.69
+ *context*	20.64	***30.03***	**24.47**	3.59	4.76
10 topics					
LocalLDA	25.98	15.94	19.76	3.59	4.41
+ *context*	23.98	21.92	22.90	3.45	4.10
STM	26.32	19.14	22.16	**4.07**	**5.88**
+ *context*	22.50	28.40	25.11	3.43	4.15

Table 3.2: ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) scores for our proposed token-level summarization approaches, compared with two baselines, supervised methods. — all use System Clusterings

Sample System Summaries

To better exemplify the summaries generated by different systems, sample output for each method is shown in Table 3.3. We see from the table that utterance-level extractive summaries (Longest DA, Prototype DA, TMM) make more coherent but still far

DRDA (1): I think if we can if we can include them at not too much extra cost, then I'd put them in, **DRDA (2)**: Uh um we we're definitely going in for voice recognition as well as LCDs, mm. **DRDA (3)**: So we've basically worked out that we're going with a simple battery, **context DA (1)**:So it's advanced integrated circuits? **context DA (2)**:the advanced chip **context DA (3)**: and a curved on one side case which is folded in on itself , um made out of rubber
Decision Abstract: It will have voice recognition, use a simple battery, and contain an advanced chip.
Longest DA & Prototype DA: Uh um we we're definitely going in for voice recognition as well as LCDs, mm. **TMM:** I think if we can if we can include them at not too much extra cost, then I'd put them in, **SVM:** cost voice recognition simple battery **CRF:** voice recognition battery **STM:** extra cost, definitely going voice recognition LCDs, simple battery **STM + context:** cost, company, advanced integrated circuits, going voice recognition, simple battery, advanced chip, curved case rubber

Table 3.3: Sample system outputs by different methods are in the third cell (methods' names are in bold). First cell contains three DRDAs supporting the decision in the second cell and three adjacent DAs of them.

from concise and compact abstracts. On the other hand, the supervised methods (SVM, CRF) that produce token-level extracts better identify the overall content of the decision abstract. Unfortunately, they require human annotation in the training phase. In comparison, the output of fine-grained topic models can cover the most useful information.

3.3 Structured Representation via Unsupervised Relation Extraction

In this section, we present an unsupervised relation extraction-based framework for focused meeting summarization. The connection between meeting summarization and

relation extraction is made in Section 3.3.1. Our proposed approach is described in Sections 3.3.2, 3.3.3, and 3.3.4. Experimental setup and results are presented in Sections 3.3.5 and 3.3.6.

3.3.1 Focused Summarization as Relation Extraction

Given the DRDAs for each meeting grouped (not necessarily correctly) according to the decisions they support, we put each cluster of DRDAs (ordered according to time within the cluster) into one "decision document". The goal will be to produce one decision abstract for each such decision document. We obtain constituent and dependency parses using the Stanford parser [Klein and Manning, 2003, de Marneffe et al., 2006]. With the corpus of constituent-parsed decision documents as the input, we will use and modify Chen et al. [2011]'s system to identify decision cue relations and decision content relations for each cluster.[4] (Section 3.3.4 will make clear how the learned decision cue relations will be used to identify decision content relations.) The salient decision content relation instances will be returned as decision summary components.

Designed for in-domain relation discovery from standard written texts (e.g., newswire), however, the Chen et al. [2011] system cannot be applied to our task directly. In our setting, for example, neither the number of relations nor the relation types is known in advance.

In the following sections, we describe the modifications needed for the spoken meeting genre and decision-focused summarization task. In particular, Chen et al. [2011] provide two mechanisms that allow for this type of tailoring: the **feature set** used to

[4]Other unsupervised relation learning methods might also be appropriate (e.g., Open IE [Banko et al., 2007]), but they generally model relations between pairs of entities and group relations only according to lexical similarity.

cluster potential relation instances into groups/types, and a set of **global constraints** that characterize the general qualities (e.g., syntactic form, prevalence, discourse behavior) of a good relation for the task.

3.3.2 The Relation Extraction Model

In this section, we describe the Chen et al. [2011] probabilistic relation learning model used for both **Decision Cue** and **Decision Content** relation extraction. The parameter estimation and constraint encoding through posterior inference are presented in Section 3.3.3.

The relation learning model takes as input clusters of DRDAs, sorted according to utterance time and concatenated into one decision document. We assume one decision will be made per document. The goal for the model is to explain how the decision documents are generated from the latent relation variables. The posterior regularization technique (Section 3.3.3) biases inference to adhere to the declarative constraints on relation instances. In general, instead of extracting relation instances strictly satisfying a set of human-written rules, features and constraints are designed to allow the model to reveal diverse relation types and to ensure that the identified relation instances are coherent and meaningful. For each decision document, we select the relation instance with highest probability for each relation type and concatenate them to form the decision summary. We restrict the eligible indicators to be a noun or verb, and eligible arguments to be a noun phrase (NP), prepositional phrase (PP) or clause introduced by "to" (S).

Given a pre-specified number of relation types K, the model employs a set of features $\phi^i(w)$ and $\phi^a(x)$ (see Section 6) to describe the indicator word w and argument constituent x. Each relation type k is associated with a set of *feature distributions* θ_k and

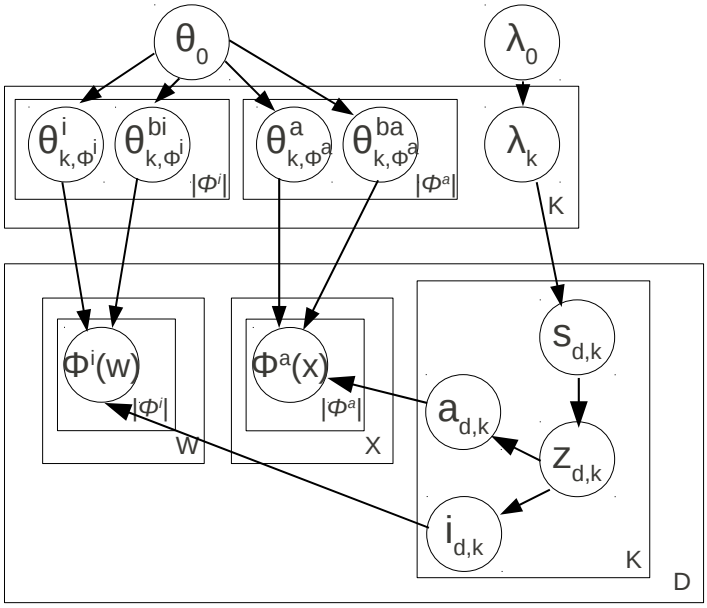


Figure 3.7: Graphical model representation for the relation learning model. D is the number of decision documents (each decision document consists of a cluster of DR-DAs). K is the number of relation types. W and X represent the number of indicators and arguments in the decision document. $|\phi^i|$ and $|\phi^a|$ are the number of features for indicator and argument.

a *location distribution* λ_k. θ_k include four parameter vectors: θ_k^i for indicator words, θ_k^{bi} for non-indicator words, θ_k^a for argument constituents, and θ_k^{ba} for non-argument constituents. Each decision document is divided into L equal-length segments and the location parameter vector λ_k describes the probability of relation k arising from each segment. The plate diagram for the model is shown in Figure 3.7. The generative process and likelihood of the model can be found in Chen et al. [2011].

3.3.3 Parameter Estimation and Inference via Posterior Regularization

In order to specify global preferences for the relation instances (e.g. the syntactic structure of the expressions), we impose inequality constraints on expectations of the posterior distributions during inference [Graca et al., 2008].

Variational inference with Constraints

Suppose we are interested in estimating the posterior distribution $p(\theta, z|x)$ of a model in general, where θ, z and x are parameters to estimate, latent variables and observations, respectively. We aim to find a distribution $q(\theta, z) \in Q$ that minimizes the KL-divergence to the true posterior

$$\mathrm{KL}(q(\theta, z)||p(\theta, z|x)) \tag{3.1}$$

A mean-field assumption is made for variational inference, where $q(\theta, z) = q(\theta)q(z)$. Then we can minimize Equation 1 by performing coordinate descent on $q(\theta)$ and $q(z)$. Now we intend to have fine-level control on the posteriors to induce meaningful semantic parts. For instance, we would like most of the extracted relation instances to satisfy a set of pre-defined syntactic patterns. As presented in Graca et al. [2008], a general way to put constraints on posterior q is through bounding expectations of given functions: $E_q[f(z)] \leq b$, where $f(z)$ is a deterministic function of z, and b is a pre-specified threshold. For instance, define $f(z)$ as a function to count the number of generated relation instances that meet the pre-defined syntactic patterns, then most of the extracted relation instances will have the desired syntactic structures. By using the mean-field assumption,

the model in Section3.3.2 is factorized as

$$
\begin{aligned}
& q(\theta, \lambda, z, i, a) = \\
& \prod_{k=1}^{K} q(\lambda_k; \hat{\lambda}_k) q(\theta_k^i; \hat{\theta}_k^i) q(\theta_k^{bi}; \hat{\theta}_k^{bi}) q(\theta_k^a; \hat{\theta}_k^a) q(\theta_k^{ba}; \hat{\theta}_k^{ba}) \\
& \times \prod_{d=1}^{D} q(z_{d,k}, i_{d,k}, a_{d,k}; \hat{c}_{d,k})
\end{aligned} \tag{3.2}
$$

The constraints are encoded in the inequalities $E_q[f(z, i, a)] \geq b$ or $E_q[f(z, i, a)] \leq b$, and affect the inference as described above. Updates for the parameters are discussed in Chen et al. [2011].

Task-Specific Constraints.

We define four types of constraints for the decision relation extraction model.

Syntactic Constraints. Syntactic constraints are widely used for information extraction (IE) systems [Snow et al., 2005, Banko and Etzioni, 2008], as it has been shown that most relations are expressed via a normalsize number of common syntactic patterns. For each relation type, we require at least 80%[5] of the induced relation instances in expectation to match one of the following syntactic patterns:

- The indicator is a verb and the argument is a noun phrase. The headword of the argument is the direct object of the indicator or the nominal subject of the indicator.

[5]Experiments show that this threshold is suitable for decision relation extraction, so we adopt it from Chen et al. [2011].

- The indicator is a verb and the argument is a prepositional phrase or a clause starting with "to". The indicator and the argument have the same parent in the constituent parsing tree.
- The indicator is a noun and is the headword of a noun phrase, and the argument is a prepositional phrase. The noun phrase with the indicator as its headword and the argument have the same parent in the constituent parsing tree.

For relation k, let $f(z_k, i_k, a_k)$ count the number of induced indicator i_k and argument a_k pairs that match one of the patterns above, and b is set to $0.8D$, where D is the number of decision documents. Then the syntactic constraint is encoded in the inequality $E_q[f(z_k, i_k, a_k)] \geq b$.

Prevalence Constraints. The prevalence constraint is enforced on the number of times a relation is instantiated, in order to guarantee that every relation has enough instantiations across the corpus and is task-relevant. Again, we require each relation to have induced instances in at least 80% of decision documents.

Occurrence Constraints. Diversity of relation types is enforced through occurrence constraints. In particular, for each decision document, we restrict each word to trigger at most two relation types as indicator and occur at most twice as part of a relation's argument in expectation. An entire span of argument constituent can appear in at most one relation type.

Discourse Constraints. The discourse constraint captures the insight that the final decision on an issue is generally made, or at least restated, at the end of the decision-related

discussion. As each decision document is divided into four equal parts, we restrict 50% of the relation instances to be from the last quarter of the decision documents.

3.3.4 Features

Basic Features
unigram (stemmed) part-of-speech (POS) constituent label (NP, VP, S/SBAR (start with "to")) dependency label
Meeting Features
Dialogue Act (DA) type speaker role topic
Structural Features [Galley, 2006, Wang and Cardie, 2011]
in an Adjacency Pair (AP)? if in an AP, AP type if in an AP, the other part is decision-related? if in an AP, the source part or target part? if in an AP and is source part, is the target positive feedback? if in an AP and is target part, is the source a question?
Semantic Features (from WordNet) [Miller, 1995]
first Synset of head word with the given POS first hypernym path for the first synset of head word
Other Features (only for Argument)
number of words (without stopwords) has capitalized word or not has proper noun or not

Table 3.4: Features for **Decision Cue** and **Decision Content** relation extraction. All features, except the last type of features, are used for both the indicator and argument. An Adjacency Pair (AP) is an important conversational analysis concept [Schegloff and Sacks, 1973]. In the AMI corpus, an AP pair consists of a source utterance and a target utterance, produced by different speakers.

Table 3.4 lists the features we use for discovering both the decision cue relations and decision content relations. We start with a collection of domain-independent BASIC FEATURES shown to be useful in relation extraction [Banko and Etzioni, 2008,

Chen et al., 2011]. Then we add MEETING FEATURES, STRUCTURAL FEATURES and SEMANTIC FEATURES that have been found to be good predictors for decision detection [Hsueh and Moore, 2007] or meeting and decision summarization [Galley, 2006, Murray and Carenini, 2008, Fernández et al., 2008, Wang and Cardie, 2011]. Features employed only for argument's are listed in the last category in Table 3.4.

After applying the features in Table 3.4 and the global constraints from Section3.3.3 in preliminary experiments, we found that the extracted relation instances are mostly derived from decision cue relations. Sample decision cue relations and instances are displayed in Table 3.5 and are not necessarily surprising: previous research [Hsueh and Moore, 2007] has observed the important role of personal pronouns, such as "we" and "I", in decision-making expressions. Notably, the decision cue is always followed by the decision content. As a result, we include two additional features (see Table 3.6) that rely on the cues to identify the decision content. Finally, we disallow content relation instances with an argument containing just a personal pronoun.

Decision Cue Relations	**Relation Instances**
Group Wrap-up / Recap	we have, we are, we say, we want
Personal Explanation	I mean, I think, I guess, I (would) say
Suggestion	do we, we (could/should) do
Final Decision	it is (gonna), it will, we will

Table 3.5: Sample **Decision Cue** relation instances. The words in parentheses are filled for illustration purposes, while they are not part of the relation instances.

Discourse Features
clause position (first, second, other)
position to the first decision cue relation if any (before, after)

Table 3.6: Additional features for **Decision Content** relation extraction, inspired by **Decision Cue** relations. Both indicator and argument use those features.

3.3.5 Experimental Setup

The Corpus. We evaluate our approach on the AMI meeting corpus [Carletta et al., 2005] that consists of 140 multi-party meetings with a wide range of annotations. The 129 scenario-driven meetings involve four participants playing different roles on a design team. Importantly, the corpus includes a short (usually one-sentence), manually constructed abstract summarizing each decision discussed in the meeting. In addition, all of the dialogue acts that support (i.e., are relevant to) each decision are annotated as such. We use the manually constructed decision abstracts as gold-standard summaries.

System Inputs. We consider two system input settings. In the **True Clusterings** setting, we use the AMI annotations to create perfect partitionings of the DRDAs for input to the summarization system; in the **System Clusterings** setting, we employ a hierarchical agglomerative clustering algorithm used for this task in our previous work [Wang and Cardie, 2011]. The clustering method groups DRDAs according to their LDA topic distribution similarity. As better approaches for DRDA clustering become available, they could be employed instead.

Evaluation Metrics. We use the widely accepted ROUGE [Lin and Hovy, 2003] evaluation measure. We adopt the ROUGE-1 and ROUGE-SU4 metrics from Hachey [2009], and also use ROUGE-2. We choose the stemming option of the ROUGE software at `http://berouge.com/` and remove stopwords from both the system and gold-standard summaries.

Training and Parameters. The Dirichlet hyperparameters are set to 0.1 for the priors. When training the model, ten random restarts are performed and each run stops when

reaching a convergence threshold (10^{-5}). Then we select the posterior with the lowest final free energy. For the parameters used in posterior constraints, we either adopt them from Chen et al. [2011] or choose them arbitrarily without tuning in the spirit of making the approach domain-independent.

We compare our decision summarization approach with (1) two unsupervised baselines, (2) the unsupervised relation-based approach of Hachey [2009], (3) two supervised methods, and (4) an upperbound derived from the gold standard decision abstracts.

The LONGEST DA **Baseline.** As in Riedhammer et al. [2010], this baseline simply selects the longest DRDA in each cluster as the summary. Thus, this baseline performs utterance-level decision summarization. Although it's possible that decision content is spread over multiple DRDAs in the cluster, this baseline and the next allow us to determine summary quality when summaries are restricted to a single utterance.

The PROTOTYPE DA **Baseline.** The second baseline selects the decision cluster prototype (i.e., the DRDA with the largest TF-IDF similarity with the cluster centroid) as the summary.

The Generic Relation Extraction (GRE) Method of Hachey [2009]. Hachey [2009] presents a generic relation extraction (GRE) for multi-document summarization. Informative sentences are extracted to form summaries instead of relation instances. Relation types are discovered by Latent Dirichlet Allocation, such that a probability is output for each relation instance given a topic (equivalent to relation). Their relation instances are named entity(NE)-mention pairs conforming to a set of pre-specified rules. For comparison, we use these same rules to select noun-mention pairs rather than NE-mention

pairs, which is better suited to meetings, which do not contain many NEs.[6]

Supervised Learning (SVMs and CRFs). We also compare our approach to two supervised learning methods — Support Vector Machines [Joachims, 1998] with RBF kernel and order-1 Conditional Random Fields [Lafferty et al., 2001b] — trained using the same features as our system (see Tables 3.4 and 3.6) to identify the important **tokens** to include in the decision abstract. Three-fold cross validation is conducted for both methods.

Upperbound. We also compute an upperbound that reflects the gap between the best possible extractive summaries and the human-written abstracts according to the ROUGE score: for each cluster of DRDAs, we select the words that also appear in the associated decision abstract.

3.3.6 Results

Table 3.7 illustrates that, using **True (DRDA) Clusterings** our method outperforms the two baselines and the generic relation extraction (GRE) based system in terms of F score in ROUGE-1 and ROUGE-SU4 with varied numbers of relations. Note that for GRE based approach, we only list out their best results for utterance-level summarization. If using the salient relation instances identified by GRE as the summaries, the ROUGE results will be significantly lower. When measured by ROUGE-2, our method

[6]Because an approximate set cover algorithm is used in GRE, one decision-related dialogue act (DRDA) is extracted each time until the summary reaches the desired length. We run two sets of experiments using this GRE system with different output summaries — one selects one entire DRDA as the final summary (as Hachey [2009] does), and another one outputs the relation instances with highest probability conditional on each relation type. We find that the first set of experiments gets better performance than the second, so we only report the best results for their system in this paper.

	True Clusterings				
	R-1			R-2	R-SU4
	PREC	REC	F1	F1	F1
Baselines					
Longest DA	34.06	31.28	32.61	12.03	13.58
Prototype DA	40.72	28.21	33.32	12.18	13.46
GRE					
5 topics	38.51	30.66	34.13	11.44	13.54
10 topics	39.39	31.01	34.69	11.28	13.42
15 topics	38.00	29.83	33.41	11.40	12.80
20 topics	37.24	30.13	33.30	10.89	12.95
Supervised Methods					
CRF	53.95	26.57	35.61	11.52	14.07
SVM	42.30	41.49	40.87	12.91	16.29
Our Method					
5 Relations	39.33	35.12	37.10	12.05	14.29
10 Relations	37.94	37.03	**37.47**	**12.20**	**14.59**
15 Relations	37.36	37.43	37.39	11.47	14.00
20 Relations	37.27	***37.64***	37.45	11.40	13.90
Upperbound	100.00	***45.05***	62.12	33.27	34.89

Table 3.7: ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) scores (multiplied by 100) for summaries produced by the baselines, GRE [Hachey, 2009]'s best results, the supervised methods, our method and an upperbound — all with perfect/true DRDA clusterings.

still have better or comparable performances than other unsupervised methods. Moreover, our system achieves F scores in between those of the supervised learning methods, performing better than the CRF in both recall and F score. The recall score for the upperbound in ROUGE-1, on the other hand, indicates that there is still a wide gap between the extractive summaries and human-written abstracts: without additional lexical information (e.g., semantic class information, ontologies) or a real language generation component, recall appears to be a bottleneck for extractive summarization methods that select content only from decision-related dialogue acts (DRDAs).

Results using the **System Clusterings** (Table 3.8) are comparable, although all of the system and baseline scores are much lower. Supervised methods get the best F scores

	System Clusterings				
	R-1			R-2	R-SU4
	PREC	REC	F1	F1	F1
Baselines					
Longest DA	17.06	11.64	13.84	2.76	3.34
Prototype DA	18.14	10.11	12.98	2.84	3.09
GRE					
5 topics	17.10	9.76	12.40	3.03	3.41
10 topics	16.28	10.03	12.35	3.00	3.36
15 topics	16.54	10.90	13.04	2.84	3.28
20 topics	17.25	8.99	11.80	2.90	3.23
Supervised Methods					
CRF	47.36	15.34	23.18	6.12	9.21
SVM	39.50	18.49	25.19	6.15	9.86
Our Method					
5 Relations	16.12	18.93	17.41	3.31	5.56
10 Relations	16.27	18.93	17.50	3.32	5.69
15 Relations	16.42	19.14	**17.68**	**3.47**	**5.75**
20 Relations	16.75	18.25	17.47	3.33	5.64

Table 3.8: ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) scores (multiplied by 100) for summaries produced by the baselines, GRE [Hachey, 2009]'s best results, the supervised methods and our method — all with system clusterings.

largely due to their high precision; but our method attains the best recall in ROUGE-1.

Discussion. To better exemplify the summaries generated by different systems, sample output for each method is shown in Table 3.9. The GRE system uses an approximate algorithm for set cover extraction, we list the first three selected DRDA in order. We see from the table that utterance-level extractive summaries (Longest DA, Prototype DA, GRE) make more coherent but still far from concise and compact abstracts. On the other hand, the supervised methods (SVM, CRF) that produce token-level extracts better identify the overall content of the decision abstract. Unfortunately, they require human annotation in the training phase; in addition, the output is ungrammatical and lacks coherence. In comparison, our system presents the decision summary in the form of phrase-based relations that provide a relatively comprehensive expression.

DRDA (1): Uh the batteries, uh we also thought about that already, **DRDA (2)**: uh will be chargeable with uh uh an option for a mount station **DRDA (3)**: Maybe it's better to to include rechargeable batteries **DRDA (4)**: We already decided that on the previous meeting. **DRDA (5)**: which you can recharge through the docking station. **DRDA (6)**: normal plain batteries you can buy at the supermarket or retail shop. Yeah.
Decision Abstract: The remote will use rechargeable batteries which recharge in a docking station.
Longest DA & Prototype DA: normal plain batteries you can buy at the supermarket or retail shop. Yeah. **GRE:** 1st: normal plain batteries you can buy at the supermarket or retail shop. Yeah. 2nd: which you can recharge through the docking station. 3rd: uh will be chargeable with uh uh an option for a mount station **SVM:** batteries include rechargeable batteries decided recharge docking station **CRF:** chargeable station rechargeable batteries **Our Method:** <option, for a mount station>, <include, rechargeable batteries>, <decided, that on the previous meeting>, <recharge, through the docking station>, <buy, normal plain batteries>

Table 3.9: Sample system outputs by different methods are in the third cell (methods' names are in bold). First cell contains the six DRDAs supporting the decision abstracted in the second cell.

3.4 Conclusion

In this chapter, we first proposed a token-level focused meeting summarization framework based on topic models and showed that modeling topic structure at the utterance-level is better at identifying relevant words and phrases than document-level models. The role of context was also studied and shown to be able to identify additional summary-worthy words.

We then presented a novel framework for focused meeting summarization based on unsupervised relation extraction. We showed that our approach outperforms unsupervised utterance-level extractive summarization baselines as well as an existing generic relation-extraction-based summarization method. Our approach also produced sum-

maries competitive with those generated by supervised methods in terms of the standard ROUGE score. Overall, we found that relation-based methods for focused summarization have potential as a technique for supporting the generation of abstractive decision summaries.

CHAPTER 4

ABSTRACT GENERATION FOR MULTI-PARTY MEETINGS

In the previous chapter, we described focused meeting summarization frameworks that generate summaries in the form of key phrases or relations. Built on the relation representation of the summary-worthy content, this chapter shows how to generate informative and fluent abstracts. It falls under our contributions on generating high quality abstractive summaries for informal noisy text.

4.1 Introduction

Meetings are a common way to collaborate, share information and exchange opinions. Consequently, automatically generated meeting summaries could be of great value to people and businesses alike by providing quick access to the essential content of past meetings. *Focused meeting summaries* have been proposed as particularly useful; in contrast to summaries of a meeting as a whole, they refer to summaries of a specific aspect of a meeting, such as the DECISIONS reached, PROBLEMS discussed, PROGRESS made or ACTION ITEMS that emerged [Carenini et al., 2011]. Our goal in this chapter is to present an automatic summarization system that can generate abstract-style focused meeting summaries to help users digest the vast amount of meeting content in an easy manner.

Existing meeting summarization systems remain largely *extractive*: their summaries are comprised exclusively of patchworks of utterances selected directly from the meetings to be summarized [Riedhammer et al., 2010, Bui et al., 2009, Xie et al., 2008]. Although relatively easy to construct, extractive approaches fall short of producing con-

C: Looking at what we've got, we we want an LCD display with a spinning wheel.
B: You have to have some push-buttons, don't you?
C: Just spinning and not scrolling, I would say.
B: I think the spinning wheel is definitely very now.
A: but since LCDs seems to be uh a definite yes,
C: We're having push-buttons on the outside
C: and then on the inside an LCD with spinning wheel,

Decision Abstract (Summary):
The remote will have push buttons outside, and an LCD and spinning wheel inside.

A: and um I'm not sure about the buttons being in the shape of fruit though.
D: Maybe make it like fruity colours or something.
C: The power button could be like a big apple or something.
D: Um like I'm just thinking bright colours.

Problem Abstract (Summary):
How to incorporate a fruit and vegetable theme into the remote.

Figure 4.1: Clips from the AMI meeting corpus [Mccowan et al., 2005]. A, B, C and D refer to distinct speakers. Also shown is the gold-standard (manual) abstract (summary) for the decision and the problem.

cise and readable summaries, largely due to the noisy, fragmented, ungrammatical and unstructured text of meeting transcripts [Liu and Liu, 2009].

In contrast, human-written meeting summaries are typically in the form of *abstracts* — distillations of the original conversation written in new language. A user study from Murray et al. [2010b] showed that people demonstrate a strong preference for abstractive summaries over extracts when the text to be summarized is conversational. Consider, for example, the two types of focused summary along with their associated dialogue snippets in Figure 4.1. We can see that extracts are likely to include unnecessary and noisy information from the meeting transcripts. On the contrary, the manually composed summaries (abstracts) are more compact and readable, and are written in a distinctly non-conversational style.

To address the limitations of extract-based summaries, we propose a complete and

fully automatic domain-independent abstract generation framework for focused meeting summarization. Following existing language generation research [Angeli et al., 2010, Konstas and Lapata, 2012], we first perform *content selection*: given the dialogue acts relevant to one element of the meeting (e.g. a single decision or problem), we train a classifier to identify summary-worthy phrases. Next, we develop an "overgenerate-and-rank" strategy [Walker et al., 2001, Heilman and Smith, 2010] for *surface realization*, which generates and ranks candidate sentences for the abstract. After redundancy reduction, the full meeting abstract can thus comprise the focused summary for each meeting element. As described in subsequent sections, the generation framework allows us to identify and reformulate the important information for the focused summary. Our contributions are as follows:

- To the best of our knowledge, our system is the first fully automatic system to generate natural language abstracts for spoken meetings.
- We present a novel template extraction algorithm, based on Multiple Sequence Alignment (MSA) [Durbin et al., 1998], to induce domain-independent templates that guide abstract generation. MSA is commonly used in bioinformatics to identify equivalent fragments of DNAs [Durbin et al., 1998] and has also been employed for learning paraphrases [Barzilay and Lee, 2003].
- Although our framework requires labeled training data for each type of focused summary (decisions, problems, etc.), we also make initial tries for domain adaptation so that our summarization method does not need human-written abstracts for each new meeting domain (e.g. faculty meetings, theater group meetings, project group meetings).

We instantiate the abstract generation framework on two corpora from disparate domains — the AMI Meeting Corpus [Mccowan et al., 2005] and ICSI Meeting Cor-

pus [Janin et al., 2003] — and produce systems to generate focused summaries with regard to four types of meeting elements: DECISIONs, PROBLEMs, ACTION ITEMSs, and PROGRESS. Automatic evaluation (using ROUGE [Lin and Hovy, 2003] and BLEU [Papineni et al., 2002]) against manually generated focused summaries shows that our summarizers uniformly and statistically significantly outperform two baseline systems as well as a state-of-the-art supervised extraction-based system. Human evaluation also indicates that the abstractive summaries produced by our systems are more linguistically appealing than those of the utterance-level extraction-based system, preferring them over summaries from the extraction-based system of comparable semantic correctness (62.3% vs. 37.7%).

Finally, we examine the generality of our model across domains for two types of focused summarization — decisions and problems — by training the summarizer on out-of-domain data (i.e. the AMI corpus for use on the ICSI meeting data, and vice versa). The resulting systems yield results comparable to those from the same system trained on in-domain data, and statistically significantly outperform supervised extractive summarization approaches trained on in-domain data.

4.2 The Framework

Our domain-independent abstract generation framework produces a summarizer that generates a grammatical abstract from a cluster of *meeting-element-related dialogue acts (DAs)* — all utterances associated with a single decision, problem, action item or progress step of interest. Note that identifying these DA clusters is a difficult task in itself [Bui et al., 2009]. Accordingly, our experiments evaluate two conditions — one in which we assume that they are perfectly identified, and one in which we identify the

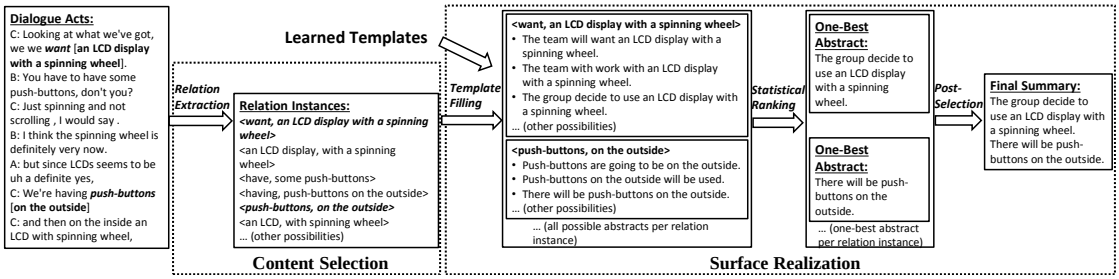


Figure 4.2: The meeting abstract generation framework. It takes as input a cluster of meeting-item-specific dialogue acts, from which one focused summary is constructed. Note that we only display *decision-related dialogue acts* — utterances associated with the decision of interest. Sample relation instances are denoted in **bold** (The indicators are further *italicized* and the arguments are in [brackets]). Summary-worthy relation instances are identified by content selection module (see Section 4.3) and then filled into the learned templates individually. A statistical ranker subsequently selects one best abstract per relation instance (see Section 4.4.2). The post-selection component reduces the redundancy and outputs the final summary (see Section 4.4.3).

clusters automatically.

The summarizer consists of two major components and is depicted in Figure 4.2. Given the DA cluster to be summarized, the *Content Selection* module identifies a set of summary-worthy *relation instances* represented as indicator-argument pairs (i.e. these constitute a finer-grained representation than DAs). The *Surface Realization* component then generates a short summary in three steps. In the first step, each relation instance is filled into templates with disparate structures that are learned automatically from the training set (*Template Filling*). A statistical ranker then selects one best abstract per relation instance (*Statistical Ranking*). Finally, selected abstracts are processed for redundancy removal in *Post-Selection*. Detailed descriptions for each individual step are provided in Sections 4.3 and 4.4.

4.3 Content Selection

Phrase-based content selection approaches have been shown to support better meeting summaries [Fernández et al., 2008]. Therefore, we chose a content selection representation of a finer granularity than an utterance: we identify *relation instances* that can both effectively detect the crucial content and incorporate enough syntactic information to facilitate the downstream surface realization.

More specifically, our relation instances are based on information extraction methods that identify a lexical *indicator* (or *trigger*) that evokes a relation of interest and then employ syntactic information, often in conjunction with semantic constraints, to find the *argument constituent*(or *target phrase*) to be extracted. Relation instances, then, are represented by **indicator-argument** pairs [Chen et al., 2011]. For example, in the DA cluster of Figure 4.2, ⟨*want, an LCD display with a spinning wheel*⟩ and ⟨*push-buttons, on the outside*⟩ are two relation instances.

Relation Instance Extraction. We adopt and extend the syntactic constraints from rel to identify all relation instances in the input utterances; the summary-worthy ones will be selected by a discriminative classifier. Constituent and dependency parses are obtained by the Stanford parser [Klein and Manning, 2003]. Both the indicator and argument take the form of constituents in the parse tree. We restrict the eligible indicator to be a noun or verb; the eligible arguments is a noun phrase (NP), prepositional phrase (PP) or adjectival phrase (ADJP). A valid indicator-argument pair should have at least one content word and satisfy one of the following constraints:

- When the indicator is a noun, the argument has to be a modifier or complement of the indicator.

- When the indicator is a verb, the argument has to be the subject or the object if it is an NP, or a modifier or complement of the indicator if it is a PP/ADJP.

We view relation extraction as a binary classification problem rather than a clustering task [Chen et al., 2011]. All relation instances can be categorized as summary-worthy or not, but only the summary-worthy ones are used for abstract generation. A discriminative classifier is trained for this purpose based on Support Vector Machines (SVMs) [Joachims, 1998] with an RBF kernel. For training data construction, we consider a relation instance to be a positive example if it shares any content word with its corresponding abstracts, and a negative example otherwise. The features used are shown in Table 4.1.

4.4 Surface Realization

In this section, we describe surface realization, which renders the relation instances into natural language abstracts. This process begins with template extraction (Section 4.4.1). Once the templates are learned, the relation instances from Section 4.3 are filled into the templates to generate an abstract (see Section 4.4.2). Redundancy handling is discussed in Section 4.4.3.

4.4.1 Template Extraction

Sentence Clustering. Template extraction starts with clustering the sentences that constitute the manually generated abstracts in the training data according to their lexical and structural similarity. From each cluster, multiple-sequence alignment techniques are employed to capture the recurring patterns.

Basic Features
number of words/content words
portion of content words/stopwords
number of content words in indicator/argument
number of content words that are also in previous DA
indicator/argument only contains stopword?
number of new nouns
Content Features
has capitalized word?
has proper noun?
TF/IDF/TFIDF min/max/average
Discourse Features
main speaker or not?
is in an adjacency pair (AP)?
is in the source/target of the AP?
number of source/target DA in the AP
is the target of the AP a positive/negative/neutral response?
is the source of the AP a question?
Syntax Features
indicator/argument constituent tag
dependency relation of indicator and argument

Table 4.1: Features for content selection. Most are adapted from previous work [Galley, 2006, Xie et al., 2008, rel]. Every basic or content feature is concatenated with the constituent tags of indicator and argument to compose a new one. Main speakers include the most talkative speaker (who has said the most words) and other speakers whose word count is more than 20% of the most talkative one [Xie et al., 2008]. Adjacency pair (AP) [Galley, 2006] is an important conversational analysis concept; each AP consists of a source utterance and a target utterance produced by different speakers.

Intuitively, desirable templates are those that can be applied in different domains to generate the same type of focused summary (e.g. decision or problem summaries). We do not want sentences to be clustered only because they describe the same domain-specific details (e.g. they are all about "data collection"), which will lead to fragmented templates that are not reusable for new domains. We therefore replace all appearances of dates, numbers, and proper names with generic labels. We also replace words that appear in both the abstract and supporting dialogue acts by a label indicating its phrase type. For any noun phrase with its head word abstracted, the whole phrase is also replaced

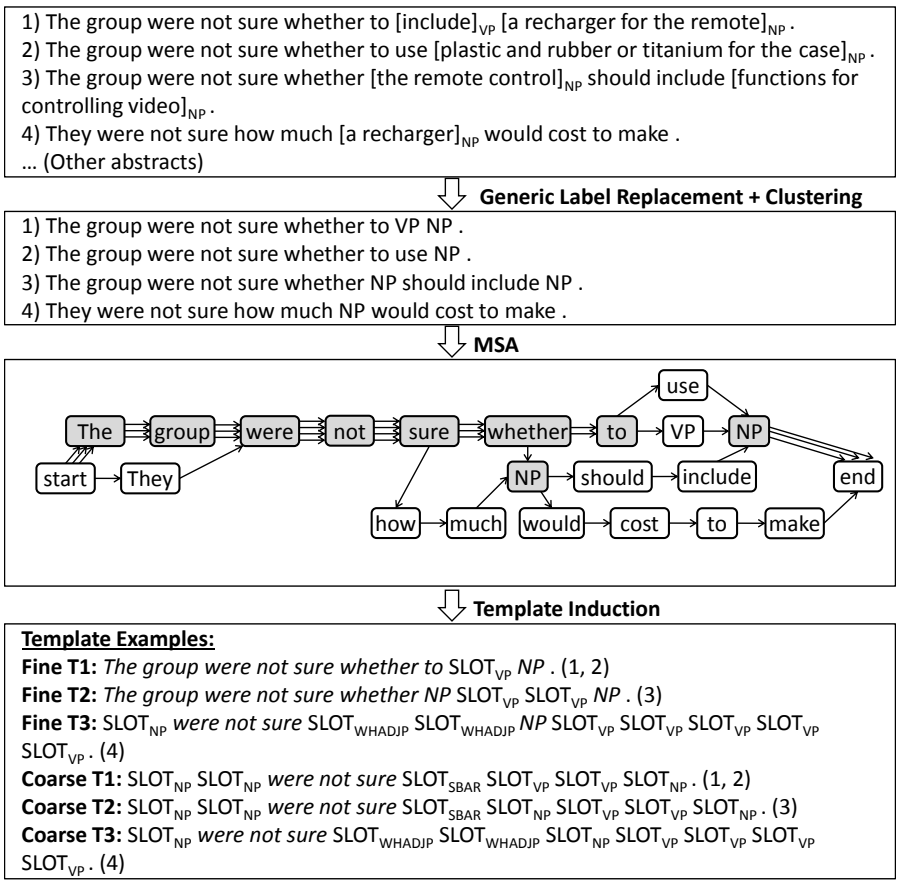


Figure 4.3: Example of template extraction by Multiple-Sequence Alignment for problem abstracts from AMI. Backbone nodes shared by at least 50% sentences are shaded. The grammatical errors exist in the original abstracts.

with "NP".

Following Barzilay and Lee [2003], we approach the sentence clustering task by hierarchical complete-link clustering with a similarity metric based on word n-gram overlap ($n = 1, 2, 3$). Clusters with fewer than three abstracts are removed[1].

Learning the Templates via MSA. For learning the structural patterns among the abstracts, *Multiple-Sequence Alignment (MSA)* is first computed for each cluster. MSA

[1]Clustering stops when the similarity between any pairwise clusters is below 5. This is applied to every type of summarization. We tune the parameter on a normalsize held-out development set by manually evaluating the induced templates. No significant change is observed within a normalsize range.

takes as input multiple sentences and one scoring function to measure the similarity between any two words. For insertions or deletions, a gap cost is also added. MSA can thus find the best way to align the sequences with insertions or deletions in accordance with the scorer. However, computing an optimal MSA is NP-complete [Wang and Jiang, 1994], thus we implement an approximate algorithm [Needleman and Wunsch, 1970] that iteratively aligns two sequences each time and treats the resulting alignment as a new sequence[2]. Figure 4.3 demonstrates an MSA computed from a sample cluster of abstracts. The MSA is represented in the form of word lattice, from which we can detect the structural similarities shared by the sentences.

To transform the resulting MSAs into templates, we need to decide whether a word in the sentence should be retained to comprise the template or abstracted. The *backbone* nodes in an MSA are identified as the ones shared by more than 50%[3] of the cluster's sentences (shaded in gray in Figure 4.3). We then create a FINE template for each sentence by abstracting the non-backbone words, i.e. replacing each of those words with a generic token (last step in Figure 4.3). We also create a COARSE template that only preserves the nodes shared by all of the cluster's sentences. By using the operations above, domain-independent patterns are thus identified and domain-specific details are removed.

Note that we do not explicitly evaluate the quality of the learned templates, which would require a significant amount of manual evaluation. Instead, they are evaluated extrinsically. We encode the templates as features [Angeli et al., 2010] that could be selected or ignored in the succeeding abstract ranking model.

[2]We adopt the scoring function for MSA from Barzilay and Lee [2003], where aligning two identical words scores 1, inserting a gap scores −0.01, and aligning two different words scores −0.5.

[3]See Barzilay and Lee [2003] for a detailed discussion about the choice of 50% according to pigeonhole principle.

4.4.2 Template Filling

An Overgenerate-and-Rank Approach. Since filling the relation instances into templates of distinct structures may result in abstracts of varying quality, we rank the abstracts based on the features of the template, the transformation conducted, and the generated abstract. This is realized by the *Overgenerate-and-Rank* strategy [Walker et al., 2001, Heilman and Smith, 2010]. It takes as input a set of relation instances (from the same cluster) $R = \{\langle ind_i, arg_i \rangle\}_{i=1}^{N}$ that are produced by content selection component, a set of templates $T = \{t_j\}_{j=1}^{M}$ that are represented as parsing trees, a transformation function F (described below), and a statistical ranker S for ranking the generated abstracts, for which we defer description later in this Section.

For each $\langle ind_i, arg_i \rangle$, the overgenerate-and-rank approach fills it into each template in T by applying F to generate all possible abstracts. Then the ranker S selects the best abstract abs_i. Post-selection is conducted on the abstracts $\{abs_i\}_{i=1}^{N}$ to form the final summary.

The transformation function F models the *constituent-level* transformations of relation instances and their mappings to the parse trees of templates. With the intuition that people will reuse the relation instances from the transcripts albeit not necessarily in their original form to write the abstracts, we consider three major types of mapping operations for the indicator or argument in the source pair, namely, *Full-Constituent Mapping*, *Sub-Constituent Mapping*, and *Removal*. *Full-Constituent Mapping* denotes that a source constituent is mapped directly to a target constituent of the template parse tree with the same tag. *Sub-Constituent Mapping* encodes more complex and flexible transformations in that a sub-constituent of the source is mapped to a target constituent with the same tag. This operation applies when the source has a tag of PP or ADJP, in which case its sub-constituent, if any, with a tag of NP, VP or ADJP can be mapped

to the target constituent with the same tag. For instance, an argument "with a spinning wheel" (PP) can be mapped to an NP in a template because it has a sub-constituent "a spinning wheel" (NP). *Removal* means a source is not mapped to any constituent in the template.

Formally, F is defined as:

$$F(\langle ind^{src}, arg^{src}\rangle, t) = \{\langle ind_k^{tran}, arg_k^{tran}, ind_k^{tar}, arg_k^{tar}\rangle\}_{k=1}^{K}$$

where $\langle ind^{src}, arg^{src}\rangle \in R$ is a relation instance (*source pair*); $t \in T$ is a template; ind_k^{tran} and arg_k^{tran} is the *transformed pair* of ind^{src} and arg^{src}; ind_k^{tar} and arg_k^{tar} are constituents in t, and they compose one *target pair* for $\langle ind^{src}, arg^{src}\rangle$. We require that ind^{src} and arg^{src} are not removed at the same time. Moreover, for valid ind_k^{tar} and arg_k^{tar}, the words subsumed by them should be all abstracted in the template, and they do not overlap in the parse tree.

To obtain the realized abstract, we traverse the parse tree of the filled template in pre-order. The words subsumed by the leaf nodes are thus collected sequentially.

Learning a Statistical Ranker. We utilize a discriminative ranker based on Support Vector Regression (SVR) [Smola and Schölkopf, 2004] to rank the generated abstracts. Given the training data that includes clusters of gold-standard summary-worthy relation instances, associated abstracts they support, and the parallel templates for each abstract, training samples for the ranker are constructed according to the transformation function F mentioned above. Each sample is represented as: $(\langle ind^{src}, arg^{src}\rangle, \langle ind_k^{tran}, arg_k^{tran}, ind_k^{tar}, arg_k^{tar}\rangle, t, a)$ where $\langle ind^{src}, arg^{src}\rangle$ is the source pair, $\langle ind_k^{tran}, arg_k^{tran}\rangle$ is the transformed pair, $\langle ind_k^{tar}, arg_k^{tar}\rangle$ is the target pair in template t, and a is the abstract parallel to t.

We first find $\langle ind_k^{tar,abs}, arg_k^{tar,abs}\rangle$, which is the corresponding constituent pair

of $\langle ind_k^{tar}, arg_k^{tar} \rangle$ in a. Then we identify the summary-worthy words subsumed by $\langle ind_k^{tran}, arg_k^{tran} \rangle$ that also appear in a. If those words are all subsumed by $\langle ind_k^{tar,abs}, arg_k^{tar,abs} \rangle$, then it is considered to be a positive sample, and a negative sample otherwise. Table 4.2 displays the features used in abstract ranking.

Basic Features
number of words in ind^{src}/arg^{src}
number of new nouns in ind^{src}/arg^{src}
ind_k^{tran}/arg_k^{tran} only has stopword?
number of new nouns in ind_k^{tran}/arg_k^{tran}
Structure Features
constituent tag of ind^{src}/arg^{src}
constituent tag of ind^{src} with constituent tag of ind^{tar}
constituent tag of arg^{src} with constituent tag of arg^{tar}
transformation of ind^{src}/arg^{src} combined with constituent tag
dependency relation of ind^{src} and arg^{src}
dependency relation of ind^{tar} and arg^{tar}
above 2 features have same value?
Template Features
template type (fine/coarse)
realized template (e.g. “the group decided to”)
number of words in template
the template has verb?
Realization Features
realization has verb?
realization starts with verb?
realization has adjacent verbs/NPs?
ind^{src} precedes/succeeds arg^{src}?
ind^{tar} precedes/succeeds arg^{tar}?
above 2 features have same value?
Language Model Features
log p_{LM}(first word in ind_k^{tran} |previous 1/2 words)
log p_{LM}(realization)
log p_{LM}(first word in arg_k^{tran} |previous 1/2 words)
log p_{LM}(realization)/length
log p_{LM}(next word | last 1/2 words in ind_k^{tran})
log p_{LM}(next word | last 1/2 words in arg_k^{tran})

Table 4.2: Features for abstracts ranking. The language model features are based on a 5-gram language model trained on Gigaword [Graff, 2003] by SRILM [Stolcke, 2002].

4.4.3 Post-Selection: Redundancy Handling.

Post-selection aims to maximize the information coverage and minimize the redundancy of the summary. Given the generated abstracts $A = \{abs_i\}_{i=1}^{N}$, we use a greedy algorithm [Lin and Bilmes, 2010] to select a subset A', where $A' \subseteq A$, to form the final summary. We define w_{ij} as the unigram similarity between abstracts abs_i and abs_j, $C(abs_i)$ as the number of words in abs_i. We employ the following objective function: $f(A, G) = \sum_{abs_i \in A \setminus G} \sum_{abs_j \in G} w_{i,j}$, $G \subseteq A$

Algorithm 2 sequentially finds an abstract with the greatest ratio of objective function gain to length, and add it to the summary if the gain is non-negative.

Input : relation instances $R = \{\langle ind_i, arg_i \rangle\}_{i=1}^{N}$, generated abstracts $A = \{abs_i\}_{i=1}^{N}$, objective function f , cost function C
Output: final abstract G

$G \leftarrow \Phi$ (empty set);
$U \leftarrow A$;
while $U \neq \Phi$ **do**
  $abs \leftarrow \arg\max_{abs_i \in U} \frac{f(A, G \cup abs_i) - f(A,G)}{C(abs_i)}$;
  if $f(A, G \cup abs) - f(A, G) \geq 0$ **then**
    $G \leftarrow G \cup abs$;
  end
  $U \leftarrow U \setminus abs$;
end

Algorithm 2: Greedy algorithm for post-selection to generate the final summary.

4.5 Experimental Setup

Corpora. Two disparate corpora are used for evaluation. The AMI meeting corpus [Mccowan et al., 2005] contains 139 scenario-driven meetings, where groups of four people participate in a series of four meetings for a fictitious project of designing

remote control. The ICSI meeting corpus [Janin et al., 2003] consists of 75 naturally occurring meetings, each of them has 4 to 10 participants. Compared to the fabricated topics in AMI, the conversations in ICSI tend to be specialized and technical, e.g. discussion about speech and language technology. We use 57 meetings in ICSI and 139 meetings in AMI that include a short (usually one-sentence), manually constructed abstract summarizing each important output for every meeting. Decision and problem summaries are annotated for both corpora. AMI has extra action item summaries, and ICSI has progress summaries. The set of dialogue acts that support each abstract are annotated as such.

System Inputs. We consider two system input settings. In the **True Clusterings** setting, we use the annotations to create perfect partitions of the DAs for input to the system; in the **System Clusterings** setting, we employ a hierarchical agglomerative clustering algorithm used for this task in Wang and Cardie [2011]. DAs are grouped according to a classifier trained beforehand.

Baselines and Comparisons. We compare our system with (1) two unsupervised baselines, (2) two supervised extractive approaches, and (3) an oracle derived from the gold standard abstracts.

Baselines. As in Riedhammer et al. [2010], the LONGEST DA in each cluster is selected as the summary. The second baseline picks the cluster prototype (i.e. the DA with the largest TF-IDF similarity with the cluster centroid) as the summary according to Wang and Cardie [2011]. Although it is possible that important content is spread over multiple DAs, both baselines allow us to determine summary quality when summaries are restricted to a single utterance.

Supervised Learning. We also compare our approach to two supervised extractive summarization methods — Support Vector Machines [Joachims, 1998] trained with the same features as our system (see Table 4.1) to identify the important **DAs** (no syntax features) [Xie et al., 2008, Sandu et al., 2010] or **tokens** [Fernández et al., 2008] to include into the summary[4].

Oracle. We compute an oracle consisting of the words from the DA cluster that also appear in the associated abstract to reflect the gap between the best possible extracts and the human abstracts.

4.6 Results

Content Selection Evaluation. We first employ ROUGE [Lin and Hovy, 2003] to evaluate the content selection component with respect to the human written abstracts. ROUGE computes the ngram overlapping between the system summaries with the reference summaries, and has been used for both text and speech summarization [Dang, 2005, Xie et al., 2008]. We report ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) that are shown to correlate with human evaluation reasonably well.

In AMI, four meetings of different functions are carried out in each group[5]. 35 meetings for "conceptual design" are randomly selected for testing. For ICSI, we reserve 12 meetings for testing.

The R-SU4 scores for each system are displayed in Figure 4.4 and show that our system uniformly outperforms the baselines and supervised systems. The learning curve

[4]We use SVM^{light} [Joachims, 1999] with RBF kernel by default parameters for SVM-based classifiers and regressor.

[5]The four types of meetings in AMI are: project kick-off (35 meetings), functional design (35 meetings), conceptual design (35 meetings), and detailed design (34 meetings).

of our system is relatively flat, which means not many training meetings are required to reach a usable performance level.

Note that the ROUGE scores are relative low when the reference summaries are human abstracts, even for evaluation among abstracts produced by different annotators [Dang, 2005]. The intrinsic difference of styles between dialogue and human abstract further lowers the scores. But the trend is still respected among the systems.

Abstract Generation Evaluation. To evaluate the full abstract generation system, the BLEU score [Papineni et al., 2002] (the precision of unigrams and bigrams with a brevity penalty) is computed with human abstracts as reference. BLEU has a fairly good agreement with human judgement and has been used to evaluate a variety of language generation systems [Angeli et al., 2010, Konstas and Lapata, 2012].

We are not aware of any existing work generating abstractive summaries for conversations. Therefore, we compare our full system against a supervised utterance-level extractive method based on SVMs along with the baselines. The BLEU scores in Figure 4.5 show that our system improves the scores consistently over the baselines and the SVM-based approach.

Domain Adaptation Evaluation. We further examine our system in domain adaptation scenarios for decision and problem summarization, where we train the system on AMI for use on ICSI, and vice versa. Table 4.3 indicates that, with both true clusterings and system clusterings, our system trained on out-of-domain data achieves comparable performance with the same system trained on in-domain data. In most experiments, it also significantly outperforms the baselines and the extract-based approaches ($p < 0.05$).

System (True Clusterings)	AMI Decision			ICSI Decision			AMI Problem			ICSI Problem		
	R-2	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*
CENTROID DA	1.3	3.0	7.7	1.8	3.5	3.8	1.0	2.7	4.2	1.0	2.3	2.8
LONGEST DA	1.6	3.3	7.0	2.8	4.7	6.5	1.0	3.0	3.6	1.2	3.4	4.6
SVM-DA (IN)	3.4	4.7	9.7	3.4	4.5	5.7	1.4	2.4	5.0	1.6	3.4	3.4
SVM-DA (OUT)	2.7	4.2	6.6	3.1	4.2	4.6	1.4	2.2	2.5	1.3	3.0	4.6
OUR SYSTEM (IN)	**4.5**	**6.2**	**11.6**	**4.9**	**7.1**	**10.0**	**3.1**	**4.8**	**7.2**	**4.0**	**5.9**	**6.0**
OUR SYSTEM (OUT)	**4.6**	**6.1**	**10.3**	**4.8**	**6.4**	*7.8*	**3.5**	**4.7**	*6.2*	*3.0*	**5.5**	*5.3*
ORACLE	7.5	12.0	22.8	9.9	14.9	20.2	6.6	11.3	18.9	6.4	12.6	13.0
System (System Clusterings)	**AMI Decision**			**ICSI Decision**			**AMI Problem**			**ICSI Problem**		
	R-2	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*
CENTROID DA	1.4	3.3	3.8	1.4	2.1	2.0	0.8	2.8	2.9	0.9	2.3	1.8
LONGEST DA	1.4	3.3	5.7	1.7	3.4	5.5	0.8	3.2	4.1	0.9	3.4	4.4
SVM-DA (IN)	2.6	4.6	10.5	3.5	6.5	7.1	1.8	3.7	4.9	1.8	4.0	4.6
SVM-DA (OUT)	3.4	5.8	10.3	2.7	4.8	6.3	2.1	3.8	4.3	1.5	3.8	3.5
OUR SYSTEM (IN)	*3.5*	*5.4*	*11.7*	**4.4**	**7.4**	**9.1**	**3.3**	**4.6**	**9.5**	*2.3*	*4.2*	**7.4**
OUR SYSTEM (OUT)	**3.9**	**6.4**	*11.4*	**4.1**	*5.1*	**8.4**	**3.6**	**5.6**	**8.9**	*1.8*	*4.0*	**6.8**
ORACLE	6.4	12.0	15.1	8.2	15.2	17.6	6.5	13.0	20.9	5.5	11.9	15.5

Table 4.3: Domain adaptation evaluation. Systems trained on out-of-domain data are denoted with "(OUT)", otherwise with "(IN)". ROUGE and BLEU scores are multiplied by 100. Our systems that statistically significantly outperform all the other approaches (except ORACLE) are in **bold** ($p < 0.05$, paired t-test). The numbers in *italics* show the significant improvement over the baselines by our systems.

Human Evaluation. We randomly select 15 decision and 15 problem DA clusters (true clusterings). We evaluate **fluency** (is the text grammatical?) and **semantic correctness** (does the summary convey the gist of the DAs in the cluster?) for OUR SYSTEM trained on IN-domain data and OUT-of-domain data, and for the utterance-level extraction system (SVM-DA) trained on in-domain data. Each cluster of DAs along with three randomly ordered summaries are presented to the judges. Five native speaking Ph.D. students (none are authors) performed the task.

We carry out an one-way Analysis of Variance which shows significant differences in score as a function of system ($p < 0.05$, paired t-test). Results in Table 4.4 demonstrate that our system summaries are significantly more compact and fluent than the extract-based method ($p < 0.05$) while semantic correctness is comparable.

The judges also **rank** the three summaries in terms of the overall quality in content,

System	Fluency		Semantic		Length
	Mean	*S.D.*	*Mean*	*S.D.*	
OUR SYSTEM (IN)	**3.67**	0.85	3.27	1.03	**23.65**
OUR SYSTEM (OUT)	3.58	0.90	3.25	1.16	**24.17**
SVM-DA (IN)	3.36	0.84	3.44	1.26	38.83

Table 4.4: Human evaluation results of **Fluency** and **Semantic** correctness for the generated abstracts. The ratings are on 1 (worst) to 5 (best) scale. The average **Length** of the abstracts for each system is also listed.

conciseness and grammaticality. An inter-rater agreement of Fleiss's $\kappa = 0.45$ (moderate agreement [Landis and Koch, 1977]) was computed. Judges selected our system as the best system in 62.3% scenarios (IN-DOMAIN: 35.6%, OUT-OF-DOMAIN: 26.7%). Sample summaries are exhibited in Figure 4.6.

4.7 Conclusion

In this chapter, we presented a domain-independent abstract generation framework for focused meeting summarization. Experimental results on two disparate meeting corpora showed that our system can uniformly outperform the state-of-the-art supervised extraction-based systems in both automatic and manual evaluation. In the domain adaptation experiments, our system also exhibited an ability to train on out-of-domain data to generate abstracts for a new target domain.

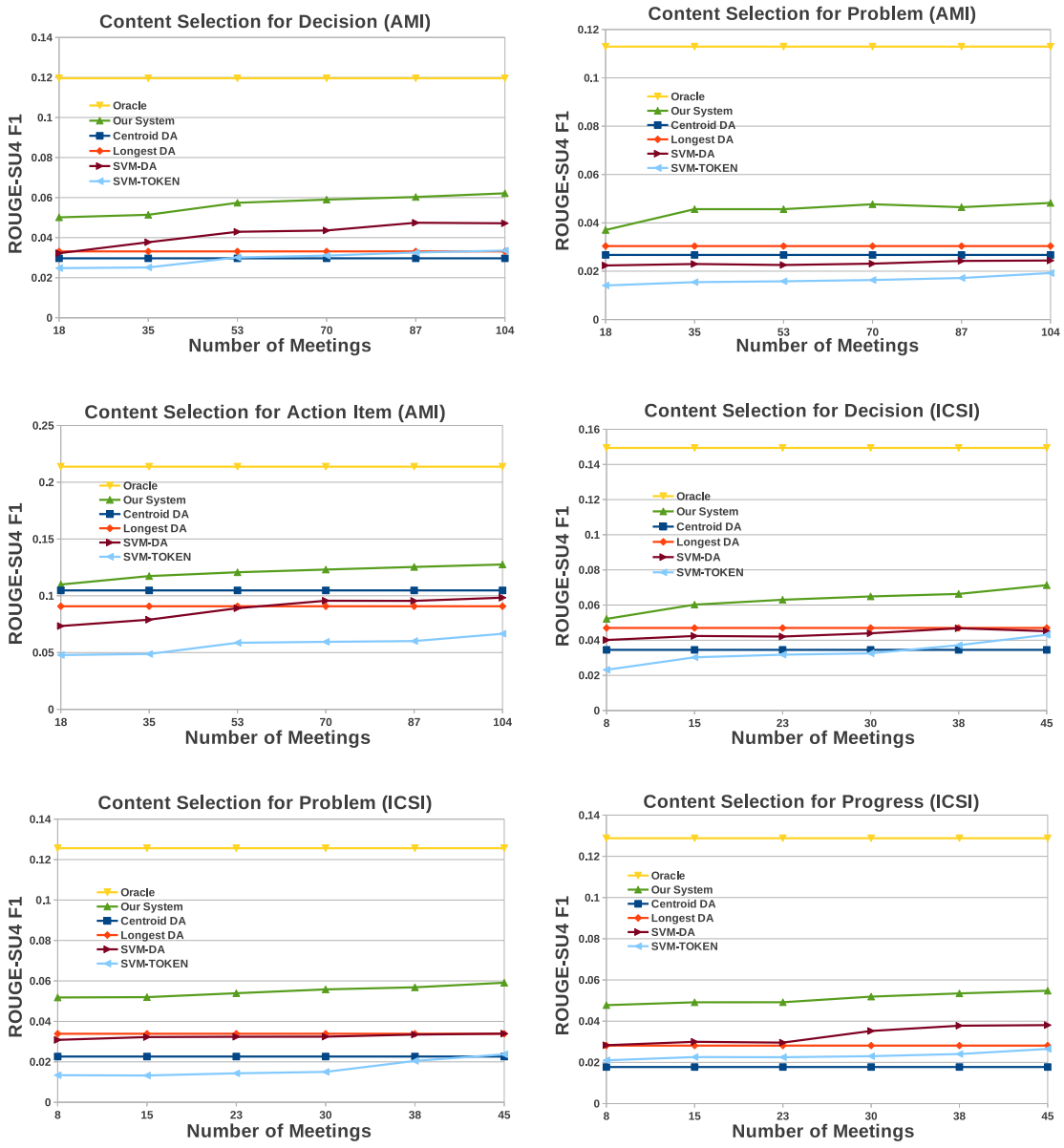


Figure 4.4: Content selection evaluation by using ROUGE-SU4 (multiplied by 100). SVM-DA and SVM-TOKEN denotes for supervised extract-based methods with SVMs on utterance- and token-level. Summaries for decision, problem, action item, and progress are generated and evaluated for AMI and ICSI (with names in parentheses). X-axis shows the number of meetings used for training.

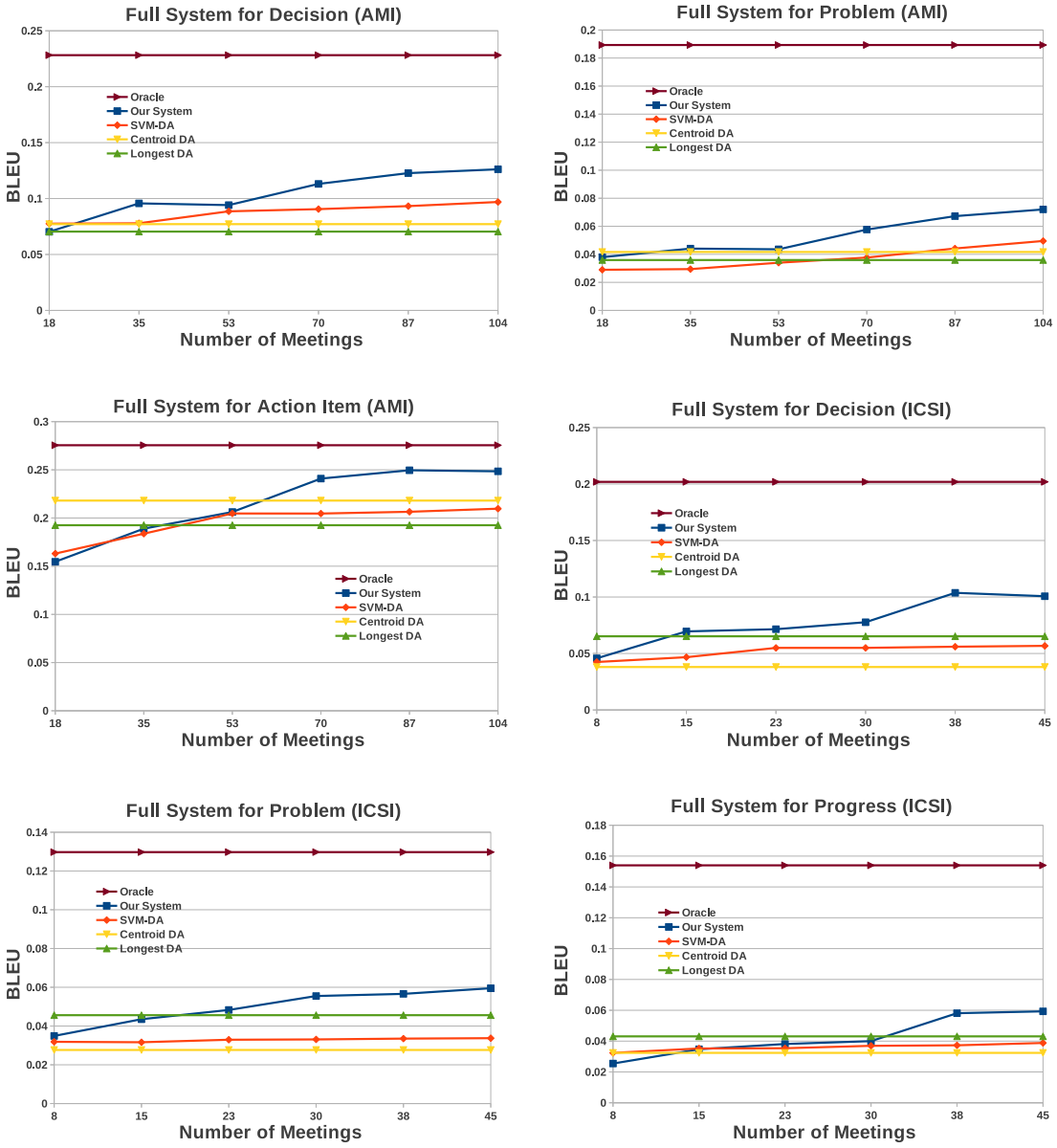


Figure 4.5: Full abstract generation system evaluation by using BLEU (multiplied by 100). SVM-DA denotes for supervised extractive methods with SVMs on utterance-level.

Decision Summary:

Human: The remote will have push buttons outside, and an LCD and spinning wheel inside.

Our System (In): The group decide to use an LCD display with a spinning wheel. There will be push-buttons on the outside.

Our System (Out): LCD display is going to be with a spinning wheel. It is necessary having push-buttons on the outside.

SVM-DA: Looking at what we've got, we we want an LCD display with a spinning wheel. Just spinning and not scrolling, I would say. I think the spinning wheel is definitely very now. We're having push-buttons on the outside

Problem Summary:

Human: How to incorporate a fruit and vegetable theme into the remote.

Our System (In): Whether to include the shape of fruit. The team had to thinking bright colors.

Our System (Out): It is unclear that the buttons being in the shape of fruit.

SVM-DA: and um Im not sure about the buttons being in the shape of fruit though.

Figure 4.6: Sample decision and problem summaries generated by various systems for examples in Figure 4.1.

CHAPTER 5

SENTENCE COMPRESSION TO MULTI-DOCUMENT SUMMARIZATION

In this chapter, we will describe a query-focused multi-document summarization system that generates concise answers for user-specified questions. This corresponds to our contribution on utilizing text summarization to address open-ended information requests.

5.1 Introduction

Another problem I have investigated is applying sentence compression techniques for multi-document summarization (MDS). The explosion of the Internet clearly warrants the development of techniques for organizing and presenting information to users in an effective way. Query-focused multi-document summarization methods have been proposed as one such technique and have attracted significant attention in recent years. The goal of query-focused MDS is to synthesize a brief (often fixed-length) and well-organized summary from a set of topic-related documents that answer a complex question or address a topic statement. The resulting summaries, in turn, can support a number of information analysis applications including open-ended question answering, recommender systems, and summarization of search engine results. As further evidence of its importance, the Document Understanding Conference (DUC) has used query-focused MDS as its main task since 2004 to foster new research on automatic summarization in the context of users' needs.

To date, most top-performing systems for multi-document summarization—whether query-specific or not—remain largely *extractive*: their summaries are comprised exclusively of sentences selected directly from the documents to be summarized [Erkan and

Radev, 2004, Haghighi and Vanderwende, 2009, Celikyilmaz and Hakkani-Tür, 2011]. Despite their simplicity, extractive approaches have some disadvantages. First, lengthy sentences that are partly relevant are either excluded from the summary or (if selected) can block the selection of other important sentences, due to summary length constraints. In addition, when people write summaries, they tend to abstract the content and seldom use entire sentences taken verbatim from the original documents. In news articles, for example, most sentences are lengthy and contain both potentially useful information for a summary as well as unnecessary details that are better omitted. Consider the following DUC query as input for an MDS system:[1] "*In what ways have stolen artworks been recovered? How often are suspects arrested or prosecuted for the thefts?*" One manually generated summary includes the following sentence but removes the bracketed words in gray:

> A man suspected of stealing a million-dollar collection of [hundreds of ancient] Nepalese and Tibetan art objects in New York [11 years ago] was arrested [Thursday at his South Los Angeles home, where he had been hiding the antiquities, police said].

In this example, the compressed sentence is relatively more succinct and readable than the original (e.g. in terms of Flesch-Kincaid Reading Ease Score [Kincaid et al., 1975]). Likewise, removing information irrelevant to the query (e.g. "11 years ago", "police said") is crucial for query-focused MDS.

Sentence compression techniques [Knight and Marcu, 2000, Clarke and Lapata, 2008] are the standard for producing a compact and grammatical version of a sentence while preserving relevance, and prior research (e.g. Lin [2003]) has demonstrated

[1] From DUC 2005, query for topic d422g.

their potential usefulness for generic document summarization. Similarly, strides have been made to incorporate sentence compression into query-focused MDS systems [Zajic et al., 2006]. Most attempts, however, fail to produce better results than those of the best systems built on pure extraction-based approaches that use no sentence compression.

In this paper we investigate the role of sentence compression techniques for query-focused MDS. We extend existing work in the area first by investigating the role of *learning-based* sentence compression techniques. In addition, we design three types of approaches to sentence-compression—*rule-based*, *sequence-based* and *tree-based*—and examine them within our compression-based framework for query-specific MDS. Our top-performing sentence compression algorithm incorporates measures of query relevance, content importance, redundancy and language quality, among others. Our tree-based methods rely on a scoring function that allows for easy and flexible tailoring of sentence compression to the summarization task, ultimately resulting in significant improvements for MDS, while at the same time remaining competitive with existing methods in terms of sentence compression, as discussed next.

We evaluate the summarization models on the standard Document Understanding Conference (DUC) 2006 and 2007 corpora [2] for query-focused MDS and find that all of our compression-based summarization models achieve statistically significantly better performance than the best DUC 2006 systems. Our best-performing system yields an 11.02 ROUGE-2 score Lin and Hovy [2003], a 8.0% improvement over the best reported score (10.2 in Davis et al. [2012]) on the DUC 2006 dataset, and an 13.49 ROUGE-2, a 5.4% improvement over the best score in DUC 2007 (12.8 in Davis et al. [2012]). We also observe substantial improvements over previous systems w.r.t. the manual Pyramid [Nenkova and Passonneau, 2004] evaluation measure (26.4 vs. 22.9 [Jagarlamudi

[2]We believe that we can easily adapt our system for othere summarization tasks (e.g. TAC-08's opinion summarization or TAC-09's update summarization) or new domains (e.g. web pages or wikipedia pages). We reserve that for future work.

et al., 2006]); human annotators furthermore rate our system-generated summaries as having less redundancy and comparable quality w.r.t. other linguistic quality metrics. With these results we believe we are the first to successfully show that sentence compression can provide statistically significant improvements over pure extraction-based approaches for query-focused MDS.

5.2 The Framework

We now present our query-focused MDS framework consisting of three steps: Sentence Ranking, Sentence Compression and Post-processing. First, sentence ranking determines the importance of each sentence given the query. Then, a sentence compressor iteratively generates the most likely succinct versions of the ranked sentences, which are cumulatively added to the summary, until a length limit is reached. Finally, the post-processing stage applies coreference resolution and sentence reordering to build the summary.

Sentence Ranking. This stage aims to rank sentences in order of relevance to the query. Unsurprisingly, ranking algorithms have been successfully applied to this task. We experimented with two of them – Support Vector Regression (SVR) [Mozer et al., 1997] and LambdaMART [Burges et al., 2007]. The former has been used previously for MDS [Ouyang et al., 2011]. LambdaMart on the other hand has shown considerable success in information retrieval tasks [Burges, 2010]; we are the first to apply it to summarization. For training, we use 40 topics (i.e. queries) from the DUC 2005 corpus [Dang, 2005] along with their manually generated abstracts. As in previous work [Shen and Li, 2011, Ouyang et al., 2011], we use the ROUGE-2 score, which measures bigram overlap between a sentence and the abstracts, as the objective for regression.

Basic Features
relative/absolute position
is among the first 1/3/5 sentences?
number of words (with/without stopwords)
number of words more than 5/10 (with/without stopwords)
Query-Relevant Features
unigram/bigram/skip bigram (at most four words apart) overlap
unigram/bigram TF/TF-IDF similarity
mention overlap
subject/object/indirect object overlap
semantic role overlap
relation overlap
Query-Independent Features
average/total unigram/bigram IDF/TF-IDF
unigram/bigram TF/TF-IDF similarity with the centroid of the cluster
average/sum of sumBasic/SumFocus Toutanova et al. [2007]
average/sum of mutual information
average/sum of number of topic signature words Lin and Hovy [2000]
basic/improved sentence scorers from Conroy et al. [2006b]
Content Features
contains verb/web link/phone number?
contains/portion of words between parentheses

Table 5.1: Sentence-level features for sentence ranking.

The features we used for sentence ranking are display in Table 5.1. Here we describe the query-relevant features, which are the most important for our summarization setting. The goal of query-relevant feature subset is to determine the similarity between the query and each candidate sentence. When computing similarity, we remove stopwords as well as the words "discuss, describe, specify, explain, identify, include, involve, note" that are adopted and extended from Conroy et al. [2006b]. Then we conduct simple query expansion based on the title of the topic and cross-document coreference resolution. Specifically, we first add the words from the topic title to the query. And for each mention in the query, we add other mentions within the set of documents that corefer with this mention. Finally, we compute two versions of the features—one based on the original query and another on the expanded one. We also derive the semantic role over-

lap and relation instance overlap between the query and each sentence. Cross-document coreference resolution, semantic role labeling and relation extraction are accomplished via the methods described in Section 5.4.

Sentence Compression. As the main focus of this paper, we propose three types of compression methods, described in detail in Section 5.3 below.

Post-processing. Post-processing performs *coreference resolution* and *sentence ordering*. We replace each pronoun with its referent unless they appear in the same sentence. For sentence ordering, each compressed sentence is assigned to the most similar (tf-idf) query sentence. Then a Chronological Ordering algorithm [Barzilay et al., 2002] sorts the sentences for each query based first on the time stamp, and then the position in the source document.

5.3 Sentence Compression

Sentence compression is typically formulated as the problem of removing secondary information from a sentence while maintaining its grammaticality and semantic structure [Knight and Marcu, 2000, McDonald, 2006, Galley and McKeown, 2007, Clarke and Lapata, 2008]. We leave other rewrite operations, such as paraphrasing and reordering, for future work. Below we describe the sentence compression approaches developed in this research: RULE-BASED COMPRESSION, SEQUENCE-BASED COMPRESSION, and TREE-BASED COMPRESSION.

Rule	Example
Header	[MOSCOW , October 19 (Xinhua) –] Russian federal troops Tuesday continued...
Relative dates	...Centers for Disease Control confirmed [Tuesday] that there was...
Intra-sentential attribution	...fueling the La Nina weather phenomenon, [the U.N. weather agency said].
Lead adverbials	[Interestingly], while the Democrats tend to talk about...
Noun appositives	Wayne County Prosecutor [John O'Hara] wanted to send a message...
Nonrestrictive relative clause	Putin, [who was born on October 7, 1952 in Leningrad], was elected in the presidential election...
Adverbial clausal modifiers (Lead sentence)	[Starting in 1998], California will require 2 per cent of a manufacturer... [Given the short time], car makers see electric vehicles as...
Within Parentheses	...to Christian home schoolers in the early 1990s [(www.homecomputermarket.com)].

Table 5.2: Linguistically-motivated rules for sentence compression. The grayed-out words in brackets are removed.

5.3.1 Rule-based Compression

Turner and Charniak [2005] have shown that applying hand-crafted rules for trimming sentences can improve both content and linguistic quality. Our rule-based approach extends existing work [Conroy et al., 2006b, Toutanova et al., 2007] to create the linguistically-motivated compression rules of Table 5.2. To avoid ill-formed output, we disallow compressions of more than 10 words by each rule.

5.3.2 Sequence-based Compression

As in McDonald [2006] and Clarke and Lapata [2008], our sequence-based compression model makes a binary "keep-or-delete" decision for each word in the sentence. In contrast, however, we view compression as a sequential tagging problem and make use of linear-chain Conditional Random Fields (CRFs) [Lafferty et al., 2001b] to select the most likely compression. We represent each sentence as a sequence of tokens, $X = x_0 x_1 \dots x_n$, and generate a sequence of labels, $Y = y_0 y_1 \dots y_n$, that encode which tokens are kept, using a BIO label format: {B-RETAIN denotes the beginning of a retained sequence, I-RETAIN indicates tokens "inside" the retained sequence, O marks tokens to

Basic Features
is first 1/3/5 tokens is last 1/3/5 tokens first letter/all letters capitalized is negation is stopword
Syntactic Tree Features
POS tag parent/grandparent label is leftmost/second leftmost child of its parent is headword in NP/VP/ADVP/ADJP chunk
Dependency Tree Features
dependency relation parent/grandparent dependency relation is the root has a depth larger than 3/5
Semantic Features
is a predicate semantic role label
Rule-Based Features
For each rule in Table 5.2 , we construct a corresponding feature to indicate whether the token is identified by the rule.

Table 5.3: Token-level features for sequence-based sentence compression.

be removed$\}$.

The CRF model is built using the features shown in Table 5.3. "Dependency Tree Features" encode the grammatical relations in which each word is involved as a dependent. For the "Syntactic Tree", "Dependency Tree" and "Rule-Based" features, we also include features for the two words that precede and the two that follow the current word. Detailed descriptions of the training data and experimental setup are in Section 5.4.

During inference, we find the maximally likely sequence Y according to a CRF with parameter θ ($Y = \arg\max_{Y'} P(Y'|X;\theta)$), while simultaneously enforcing the rules of Table 5.2 to reduce the hypothesis space and encourage grammatical compression. To do this, we encode these rules as features for each token, and whenever these feature

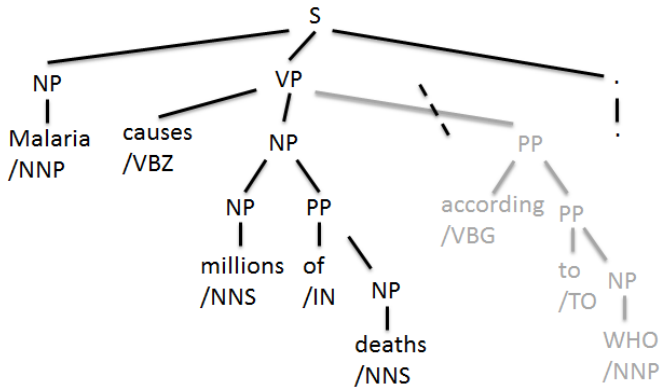


Figure 5.1: Diagram of tree-based compression. The nodes to be dropped are grayed out. In this example, the root of the gray subtree (a "PP") would be labeled REMOVE. Its siblings and parent are labeled RETAIN and PARTIAL, respectively. The trimmed tree is realized as "*Malaria causes millions of deaths.*"

functions fire, we restrict the possible label for that token to "O".

5.3.3 Tree-based Compression

Our tree-based compression methods are in line with syntax-driven approaches [Galley and McKeown, 2007], where operations are carried out on parse tree constituents. Unlike previous work [Knight and Marcu, 2000, Galley and McKeown, 2007], we do not produce a new parse tree, but focus on learning to identify the proper set of constituents to be removed. In particular, when a node is dropped from the tree, all words it subsumes will be deleted from the sentence.

Formally, given a parse tree T of the sentence to be compressed and a tree traversal algorithm, T can be presented as a list of ordered constituent nodes, $T = t_0 t_1 \dots t_m$. Our objective is to find a set of labels, $L = l_0 l_1 \dots l_m$, where $l_i \in \{$RETAIN, REMOVE, PARTIAL$\}$. RETAIN (RET) and REMOVE (REM) denote whether the node t_i is retained or removed. PARTIAL (PAR) means t_i is partly removed, i.e. at least one child subtree of t_i is dropped.

Labels are identified, in order, according to the tree traversal algorithm. Every node label needs to be *compatible* with the labeling history: given a node t_i, and a set of labels $l_0 \dots l_{i-1}$ predicted for nodes $t_0 \dots t_{i-1}$, l_i =RET or l_i =REM is *compatible* with the history when all children of t_i are labeled as RET or REM, respectively; l_i =PAR is *compatible* when t_i has at least two descendents t_j and t_k ($j < i$ and $k < i$), one of which is RETained and the other, REMoved. As such, the root of the gray subtree in Figure 5.1 is labeled as REM; its left siblings as RET; its parent as PAR.

As the space of possible compressions is exponential in the number of leaves in the parse tree, instead of looking for the globally optimal solution, we use beam search to find a set of highly likely compressions and employ a language model trained on a large corpus for evaluation.

A Beam Search Decoder. The beam search decoder (see Algorithm 3) takes as input the sentence's parse tree $T = t_0 t_1 \dots t_m$, an ordering O for traversing T (e.g. postorder) as a sequence of nodes in T, the set L of possible node labels, a scoring function S for evaluating each sentence compression hypothesis, and a beam size N. Specifically, O is a permutation on the set $\{0, 1, \dots, m\}$—each element an index onto T. Following O, T is re-ordered as $t_{O_0} t_{O_1} \dots t_{O_m}$, and the decoder considers each ordered constituent t_{O_i} in turn. In iteration i, all existing sentence compression hypotheses are expanded by one node, t_{O_i}, labeling it with **all** *compatible* labels. The new hypotheses (usually sub-sentences) are ranked by the scorer S and the top N are preserved to be extended in the next iteration. See Figure 5.2 for an example.

Our BASIC *Tree-based Compression* instantiates the beam search decoder with postorder traversal and a hypothesis scorer that takes a possible sentence compression—a sequence of nodes (e.g. $t_{O_0} \dots t_{O_k}$) and their labels (e.g. $l_{O_0} \dots l_{O_k}$)—and returns

```
Input  : parse tree T, ordering O = O_0 O_1 ... O_m, L ={RET, REM, PAR},
         hypothesis scorer S, beam size N
Output: N best compressions
stack ← Φ (empty set);
foreach node t_{O_i} in T = t_{O_0} ... t_{O_m} do
    if i == 0 (first node visited) then
        foreach label l_{O_0} in L do
            newHypothesis h' ← [l_{O_0}];
            put h' into Stack;
        end
    else
        newStack ← Φ (empty set);
        foreach hypothesis h in stack do
            foreach label l_{O_i} in L do
                if l_{O_i} is compatible then
                    newHypothesis h' ← h + [l_{O_i}];
                    put h' into newStack;
                end
            end
        end
        stack ← newStack;
    end
    Apply S to sort hypotheses in stack in descending order;
    Keep the N best hypotheses in stack;
end
```

Algorithm 3: Beam search decoder for tree-based sentence compression.

$\sum_{j=1}^{k} \log P(l_{O_j}|t_{O_j})$ (denoted later as $Score_{Basic}$). The probability is estimated by a Maximum Entropy classifier [Berger et al., 1996] trained at the constituent level using the features in Table 5.4. We also apply the rules of Table 5.2 during the decoding process. Concretely, if the words subsumed by a node are identified by any rule, we only consider REM as the node's label.

Given the N-best compressions from the decoder, we evaluate the yield of the trimmed trees using a language model trained on the Gigaword [Graff, 2003] corpus and return the compression with the highest probability. Thus, the decoder is quite flexible — its learned scoring function allows us to incorporate features salient for sentence com-

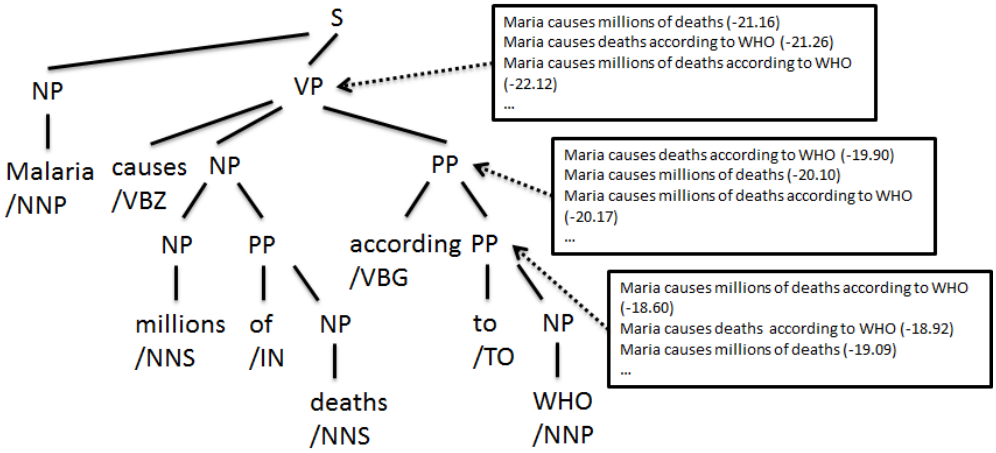


Figure 5.2: Example of beam search decoding. For postorder traversal, the three nodes are visited in a bottom-up order. The associated compression hypotheses (boxed) are ranked based on the scores in parentheses. Beam scores for other nodes are omitted.

pression while its language model guarantees the linguistic quality of the compressed string. In the sections below we consider additional improvements.

Improving Beam Search CONTEXT-*aware search* is based on the intuition that predictions on preceding context can be leveraged to facilitate the prediction of the current node. For example, parent nodes with children that have all been removed (retained) should have a label of REM (RET). In light of this, we encode these contextual predictions as additional features of S, that is, ALL-CHILDREN-REMOVED/RETAINED, ANY-LEFT-SIBLING-REMOVED/RETAINED/PARTLY_REMOVED, LABEL-OF-LEFT-SIBLING/HEAD-NODE.

HEAD-*driven search* modifies the BASIC postorder tree traversal by visiting the head node first at each level, leaving other orders unchanged. In a nutshell, if the head node is dropped, then its modifiers need not be preserved. We adopt the same features as CONTEXT-aware search, but remove those involving left siblings. We also add one more feature: LABEL-OF-THE-HEAD-NODE-IT-MODIFIES.

Basic Features
projection falls within first 1/3/5 tokens ∗
projection falls within last 1/3/5 tokens∗
contain first 1/3/5 tokens ∗
contain last 1/3/5 tokens ∗
number of words larger than 5/10 ∗
is leave node ∗
is the root of the parsing tree ∗
contain any word with first letter/all letters capitalized
contain negation
contain stopwords
Syntactic Tree Features
constituent label
parent/grandparent/left sibling/right sibling label
is leftmost/second leftmost child of its parent
is head node of its parent
its head node's constituent label
has a depth larger than 3/5/10
Dependency Tree Features
dependency relation of the head node †
dependency relation of the parent/grandparent's head node †
contain the root of the dependency tree †
has a depth larger than 3/5 †
Semantic Features
whether the head node contains a predicate
the semantic roles of the head node
Rule-Based Features
For each rule in Table 5.2 , we construct a corresponding feature to indicate whether the words in the node are identified by the rule.

Table 5.4: Constituent-level features for tree-based sentence compression. ∗ or † denote features that are concatenated with every Syntactic Tree feature to compose a new one.

Task-Specific Sentence Compression The current scorer $Score_{Basic}$ is still fairly naive in that it focuses only on features of the sentence to be compressed. *However extra-sentential knowledge can also be important for query-focused MDS*. For example, information regarding relevance to the query might lead the decoder to produce compressions better suited for the summary. Towards this goal, we construct a compression scoring function—the *multi-scorer* (MULTI)—that allows the incorporation of multi-

ple task-specific scorers. Given a hypothesis at any stage of decoding, which yields a sequence of words $W = w_0 w_1 ... w_j$, we propose the following component scorers.

Query Relevance. Query information ought to guide the compressor to identify the relevant content. The query Q is expanded as described in Section 5.2. Let $|W \cap Q|$ denote the number of unique overlapping words between W and Q, then $score_q = |W \cap Q|/|W|$.

Importance. A query-independent importance score is defined as the average SumBasic [Toutanova et al., 2007] value in W, i.e. $score_{im} = \sum_{i=1}^{j} SumBasic(w_i)/|W|$.

Language Model. Language models are widely used in many NLP problems, such as machine translation, or speech recognition. We let $score_{lm}$ be the probability of W computed by a language model.

Cross-Sentence Redundancy. To encourage diversified content, we define a redundancy score to discount replicated content: $score_{red} = 1 - |W \cap C|/|W|$, where C is the words already selected for the summary [3].

The *multi-scorer* is defined as a linear combination of the component scorer. Let $\vec{\alpha} = (\alpha_0, \ldots, \alpha_4)$, $0 \leq \alpha_i \leq 1$, $\overrightarrow{score} = (score_{Basic}, score_q, score_{im}, score_{lm}, score_{red})$,

$$S = score_{multi} = \vec{\alpha} \cdot \overrightarrow{score} \tag{5.1}$$

The parameters $\vec{\alpha}$ are tuned on a held-out tuning set by grid search. We linearly normalize the score of each metric, where the minimum and maximum values are estimated from the tuning data.

[3] A sentence will not be considered, if more than 80% of the content words have been covered by the existing summary.

5.4 Experimental Setup

We evaluate our methods on the DUC 2005, 2006 and 2007 datasets Dang [2005], duc, Dang [2007], each of which is a collection of newswire articles. 50 complex queries (topics) are provided for DUC 2005 and 2006, 35 are collected for DUC 2007 main task. Relevant documents for each query are provided along with 4 to 9 human MDS abstracts. The task is to generate a summary within 250 words to address the query. We split DUC 2005 into two parts: 40 topics to train the sentence ranking models, and 10 for ranking algorithm selection and parameter tuning for the multi-scorer. DUC 2006 and DUC 2007 are reserved as held out test sets.

Sentence Compression. The dataset from [Clarke and Lapata, 2008] is used to train the CRF and MaxEnt classifiers (Section 5.3). It includes 82 newswire articles with one manually produced compression aligned to each sentence.

Preprocessing. Documents are processed by a full NLP pipeline, including token and sentence segmentation, parsing, semantic role labeling, and an information extraction pipeline consisting of mention detection, NP coreference, cross-document resolution, and relation detection [Florian et al., 2004, Luo et al., 2004, Luo and Zitouni, 2005].

Learning for Sentence Ranking and Compression. We use Weka [Hall et al., 2009] to train a support vector regressor and experiment with various rankers in RankLib [Dang, 2011][4]. As LambdaMART has an edge over other rankers on the held-out dataset, we selected it to produce ranked sentences for further processing. For sequence-based compression using CRFs, we employ Mallet [McCallum, 2002] and integrate the Table 5.2 rules during inference. NLTK [Bird et al., 2009] MaxEnt classifiers are used for tree-

[4]Default parameters are used. If an algorithm needs a validation set, we use 10 out of 40 topics.

based compression. Beam size is fixed at 2000.[5] Sentence compressions are evaluated by a 5-gram language model trained on Gigaword [Graff, 2003] by SRILM [Stolcke, 2002].

5.5 Results

5.5.1 Automatic Evaluation

The results in Table 5.5 use the official ROUGE software with standard options[6] and report ROUGE-2 (R-2) (measures bigram overlap) and ROUGE-SU4 (R-SU4) (measures unigram and skip-bigram separated by up to four words).

We compare our sentence-compression-based methods to the best performing systems based on ROUGE in DUC 2006 and 2007 [Jagarlamudi et al., 2006, Pingali et al., 2007], system by Davis et al. [2012] that report the best R-2 score on DUC 2006 and 2007 thus far, and to the purely extractive methods of SVR and LambdaMART.

Our sentence-compression-based systems (marked with †) show statistically significant improvements over pure extractive summarization for both R-2 and R-SU4 (paired t-test, $p < 0.01$). This means our systems can effectively remove redundancy within the summary through compression. Furthermore, our HEAD-driven beam search method with MULTI-scorer beats all systems on DUC 2006[7] and all systems on DUC 2007 except the best system in terms of R-2 ($p < 0.01$). Its R-SU4 score is also significantly ($p < 0.01$) better than extractive methods, rule-based and sequence-based compression

[5] We looked at various beam sizes on the heldout data, and observed that the performance peaks around this value.

[6] ROUGE-1.5.5.pl -n 4 -w 1.2 -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -a -d

[7] The system output from Davis et al. [2012] is not available, so significance tests are not conducted on it.

	DUC 2006			DUC 2007		
System	**C Rate**	**R-2**	**R-SU4**	**C Rate**	**R-2**	**R-SU4**
Best DUC system	–	9.56	15.53	–	12.62	17.90
Davis et al. [2012]	–	10.2	15.2	–	12.8	17.5
SVR	100%	7.78	13.02	100%	9.53	14.69
LambdaMART	100%	9.84	14.63	100%	12.34	15.62
Rule-based	78.99%	10.62 $*$†	15.73 †	78.11%	13.18†	18.15†
Sequence	76.34%	10.49 †	15.60 †	77.20%	13.25†	18.23†
Tree (BASIC + $Score_{Basic}$)	70.48%	10.49 †	15.86 †	69.27%	13.00†	18.29†
Tree (CONTEXT + $Score_{Basic}$)	65.21%	10.55 $*$†	16.10 †	63.44%	12.75	18.07†
Tree (HEAD + $Score_{Basic}$)	66.70%	10.66 $*$†	16.18 †	65.05%	12.93	18.15†
Tree (HEAD + MULTI)	70.20%	**11.02** $*$†	**16.25** †	73.40%	**13.49**†	**18.46**†

Table 5.5: Query-focused MDS performance comparison: C Rate or *compression rate* is the proportion of words preserved. R-2 (ROUGE-2) and R-SU4 (ROUGE-SU4) scores are multiplied by 100. "–" indicates that data is unavailable. BASIC, CONTEXT and HEAD represent the basic beam search decoder, context-aware and head-driven search extensions respectively. $Score_{Basic}$ and MULTI refer to the type of scorer used. Statistically significant improvements ($p < 0.01$) over the best system in DUC 06 and 07 are marked with $*$. † indicates statistical significance ($p < 0.01$) over extractive approaches (SVR or LambdaMART). HEAD + MULTI outperforms all the other extract- and compression-based systems in R-2.

methods on both DUC 2006 and 2007. Moreover, our systems with learning-based compression have considerable compression rates, indicating their capability to remove superfluous words as well as improve summary quality.

5.5.2 Human Evaluation

The Pyramid [Nenkova and Passonneau, 2004] evaluation was developed to manually assess how many relevant facts or Summarization Content Units (SCUs) are captured by system summaries. We ask a professional annotator (who is not one of the authors, is highly experienced in annotating for various NLP tasks, and is fluent in English) to carry out a Pyramid evaluation on 10 randomly selected topics from the DUC 2006 task with gold-standard SCU annotation in abstracts. The Pyramid score (see Table 5.6) is

System	Pyr	Gra	Non-Red	Ref	Foc	Coh
Best DUC system (ROUGE)	22.9±8.2	3.5±0.9	3.5±1.0	3.5±1.1	3.6±1.0	2.9±1.1
Best DUC system (LQ)	–	4.0±0.8	4.2±0.7	3.8±0.7	3.6±0.9	3.4±0.9
Our System	**26.4**±10.3	3.0±0.9	4.0±1.1	3.6±1.0	3.4±0.9	2.8±1.0

Table 5.6: Human evaluation on our multi-scorer based system, Jagarlamudi et al. [2006] (Best DUC system (ROUGE)), and Lacatusu et al. [2006] (Best DUC system (LQ)). Our system can synthesize more relevant content according to Pyramid (×100). We also examine linguistic quality (LQ) in Grammaticality (Gra), Non-redundancy (Non-Red), Referential clarity (Ref), Focus (Foc), and Structure and Coherence (Coh) like [Dang, 2005], each rated from 1 (very poor) to 5 (very good). Our system has better non-redundancy than Jagarlamudi et al. [2006] and is comparable to Jagarlamudi et al. [2006] and Lacatusu et al. [2006] in other metrics except grammaticality.

re-calculated for the system with best ROUGE scores in DUC 2006 [Jagarlamudi et al., 2006] along with our system by the same annotator to make a meaningful comparison.

We further evaluate the linguistic quality (LQ) of the summaries for the same 10 topics in accordance with the measurement in [Dang, 2005]. Four native speakers who are undergraduate students in computer science (none are authors) performed the task. We compare our system based on HEAD-driven beam search with MULTI-scorer to the best systems in DUC 2006 achieving top ROUGE scores [Jagarlamudi et al., 2006] (Best DUC system (ROUGE)) and top linguistic quality scores [Lacatusu et al., 2006] (Best DUC system (LQ))[8]. The average score and standard deviation for each metric is displayed in Table 5.6. Our system achieves a higher Pyramid score, an indication that it captures more of the salient facts. We also attain better non-redundancy than Jagarlamudi et al. [2006], meaning that human raters perceive less replicative content in our summaries. Scores for other metrics are comparable to Jagarlamudi et al. [2006] and Lacatusu et al. [2006], which either uses minimal non-learning-based compression rules or is a pure extractive system. However, our compression system sometimes generates less grammatical sentences, and those are mostly due to parsing errors. For example,

[8][Lacatusu et al., 2006] obtain the best scores in three linguistic quality metrics (i.e. grammaticality, focus, structure and coherence), and overall responsiveness on DUC 2006.

parsing a clause starting with a past tense verb as an adverbial clausal modifier can lead to an ill-formed compression. Those issues can be addressed by analyzing k-best parse trees and we leave it in the future work. A sample summary from our multi-scorer based system is in Figure 5.3.

Topic *D*0626*H*: How were the bombings of the US embassies in Kenya and Tanzania conducted? What terrorist groups and individuals were responsible? How and where were the attacks planned?

WASHINGTON, August 13 (Xinhua) – President Bill Clinton Thursday condemned terrorist bomb attacks at U.S. embassies in Kenya and Tanzania and vowed to find the bombers and bring them to justice. Clinton met with his top aides Wednesday in the White House to assess the situation following the twin bombings at U.S. embassies in Kenya and Tanzania, which have killed more than 250 people and injured over 5,000, most of them Kenyans and Tanzanians. Local sources said the plan to bomb U.S. embassies in Kenya and Tanzania took three months to complete and bombers destined for Kenya were dispatched through Somali and Rwanda. FBI Director Louis Freeh, Attorney General Janet Reno and other senior U.S. government officials will hold a news conference at 1 p.m. EDT (1700GMT) at FBI headquarters in Washington "to announce developments in the investigation of the bombings of the U.S. embassies in Kenya and Tanzania," the FBI said in a statement. ... DAR ES SALAAM, Tanzania (AP) _ Investigators in the Aug. 7 bombing of the American Embassy in Tanzania said Saturday they had made "extraordinary discoveries," having determined what the bomb was made of and who carried it to the embassy. ... But authorities in Kenya and diplomats said that one man, Mustafa Mahmoud Said Ahmed, was interrogated in the fall about terrorist activities in Kenya, including a plot to bomb the American Embassy, and that he was wanted in Egypt for terrorist activities. DAR ES SALAAM, Tanzania (AP) _ An Egyptian formally charged in the Aug. 7 bombing of the U.S. Embassy in Dar es Salaam has been providing investigators with a lot of information but has been less than forthcoming about his role in the blast, sources close to the investigation say. Two suspects have been indicted in a U.S. federal court in New York in connection with the Nairobi bombing, a third is a fugitive, a fourth is awaiting extradition in Germany and a fifth has been charged in New York with lying to investigations about his relationship to Saudi exile Osama bin Laden...

Figure 5.3: Part of the summary generated by the multi-scorer based summarizer for topic *D*0626*H* (DUC 2006). Grayed out words are removed. Query-irrelevant phrases, such as temporal information or source of the news, have been removed.

5.5.3 Sentence Compression Evaluation

We also evaluate sentence compression separately on Clarke and Lapata [2008], adopting the same partitions as Martins and Smith [2009], i.e. 1,188 sentences for training and 441 for testing. Our compression models are compared with Hedge Trimmer [Dorr et al., 2003], a discriminative model proposed by McDonald [2006] and a dependency-tree based compressor Martins and Smith [2009][9]. We adopt the metrics in Martins and Smith [2009] to measure the unigram-level macro precision, recall, and F1-measure with respect to human annotated compression. In addition, we also compute the F1 scores of grammatical relations which are annotated by RASP [Briscoe and Carroll, 2002] according to Clarke and Lapata [2008].

In Table 5.7, our context-aware and head-driven tree-based compression systems show statistically significantly ($p < 0.01$) higher precisions (**Uni-Prec**) than all the other systems, without decreasing the recalls (**Uni-Rec**) significantly ($p > 0.05$) based on a paired t-test. Unigram F1 scores (**Uni-F1**) in italics indicate that the corresponding systems are not statistically distinguishable ($p > 0.05$). For grammatical relation evaluation, our head-driven tree-based system obtains statistically significantly ($p < 0.01$) better F1 score (**Rel-F1** than all the other systems except the rule-based system).

5.6 Conclusion

In this chapter, we presented a framework for query-focused multi-document summarization based on sentence compression. We proposed three types of compression approaches. Our tree-based compression method can easily incorporate measures of query relevance, content importance, redundancy and language quality into the compression

[9]Thanks to André F.T. Martins for system outputs.

System	C Rate	Uni-Prec	Uni-Rec	Uni-F1	Rel-F1
HedgeTrimmer	57.64%	0.72	0.65	0.64	0.50
McDonald (2006)	70.95%	0.77	0.78	*0.77*	0.55
Martins and Smith [2009]	71.35%	0.77	0.78	*0.77*	0.56
Rule-based	87.65%	0.74	0.91	0.80	0.63
Sequence	70.79%	0.77	0.80	*0.76*	0.58
Tree (BASIC)	69.65%	0.77	0.79	0.75	0.56
Tree (CONTEXT)	67.01%	**0.79**	0.78	*0.76*	0.57
Tree (HEAD)	68.06%	**0.79**	0.80	*0.77*	**0.59**

Table 5.7: Sentence compression comparison. The true c rate is 69.06% for the test set. Tree-based approaches all use single-scorer. Our context-aware and head-driven tree-based approaches outperform all the other systems significantly ($p < 0.01$) in precision (**Uni-Prec**) without sacrificing the recalls (i.e. there is no statistically significant difference between our models and McDonald (2006) / M & S (2009) with $p > 0.05$). *Italicized* numbers for unigram F1 (**Uni-F1**) are statistically indistinguishable ($p > 0.05$). Our head-driven tree-based approach also produces significantly better grammatical relations F1 scores (**Rel-F1**) than all the other systems except the rule-based method ($p < 0.01$).

process. By testing on a standard dataset using the automatic metric ROUGE, our models showed substantial improvement over pure extraction-based methods and state-of-the-art systems. Our best system also yielded better results for human evaluation based on Pyramid and achieves comparable linguistic quality scores.

CHAPTER 6

SUMMARIZING OPINION FROM SOCIAL MEDIA

We addressed the task of query-focused multi-document summarization in the last chapter. Here we focus on a specific type of query — complex opinion questions, which also falls under contributions on addressing open-ended information requests.

6.1 Introduction

Social media forums, such as social networks, blogs, newsgroups, and community question answering (QA), offer avenues for people to express their opinions as well collect other people's thoughts on topics as diverse as health, politics and software [Liu et al., 2008]. However, digesting the large amount of information in long threads on newsgroups, or even knowing which threads to pay attention to, can be overwhelming. A text-based summary that highlights the diversity of opinions on a given topic can lighten this information overload. In this work, we design a submodular function-based framework for opinion summarization on community question answering and blog data.

Opinion summarization has previously been applied to restricted domains, such as product reviews [Hu and Liu, 2004, Lerman et al., 2009] and news [Stoyanov and Cardie, 2006], where the output summary is either presented in a structured way with respect to each aspect of the product or organized along contrastive viewpoints. Unlike those works, we address user generated online data: community QA and blogs. These forums use a substantially less formal language than news articles, and at the same time address a much broader spectrum of topics than product reviews. As a result, they present new challenges for automatic summarization. For example, Figure 6.1 illustrates

Question: What is the long term effect of piracy on the music and film industry?

Best Answer: Rising costs for movies and music. ... If they sell less, they need to raise the price to make up for what they lost. The other thing will be music and movies with less quality. ...

Other Answers:

Ans1: Its bad... really bad. (Just watch this movie and you will find out ... Piracy causes rappers to appear on your computer).

Ans2: By removing the profitability of music & film companies, piracy takes away their motivation to produce new music & movies. If they can't protect their copyrights, they can't continue to do business. ...

Ans3: Bad news for the for the person who enjoys going to see a movie on the big screen as the cost will go through the roof. As far as quality that will get worse to because alot of the good directors and good bands are no longer going to make the movies or music.

Ans4: *It is forcing them to rework their business model, which is a good thing.* In short, I don't think the music industry in particular will ever enjoy the huge profits of the 90's. ...

Ans5: It's one of those things that really depends. I hate when people who have billions of dollars whine about not having more money. But it's also like the person put the effort into it and they aren't getting paid. It's a big gray area...

Ans6: Please-People in those businesses make millions of dollars as it is!! I don't think piracy hurts them at all!!!

Figure 6.1: Example discussion on Yahoo! Answers. Besides the best answer, other answers also contain relevant information (in *italics*). For example, the sentence in blue has a contrasting viewpoint compared to the other answers.

a sample question from Yahoo! Answers[1] along with the answers from different users. The question receives more than one answer, and one of them is selected as the "best answer" by the asker or other participants. In general, answers from other users also provide relevant information. While community QA successfully pools rich knowledge from the wisdom of the crowd, users might need to seine through numerous posts to extract the information they need. Hence, it would be beneficial to summarize answers automatically and present the summaries to users who ask similar questions in the future. In this work, we aim to return a summary that encapsulates different perspectives for a given opinion question and a set of relevant answers or documents.

[1] `http://answers.yahoo.com/`

In our work we assume that there is a central topic (or query) on which a user is seeking diverse opinions. We predict query-relevance through automatically learned statistical rankers. Our ranking function not only aims to find sentences that are on the topic of the query but also ones that are "opinionated" through the use of several features that indicate subjectivity and sentiment. The relevance score is encoded in a submodular function. Diversity is accounted for by a dispersion function that maximizes the pairwise distance between the pairs of sentences selected.

Our chief contributions are:

- We develop a submodular function-based framework for query-focused opinion summarization. To the best of our knowledge, this is the first time that submodular functions have been used to support opinion summarization. We test our framework on two tasks: summarizing opinionated sentences in community QA (Yahoo! Answers) and blogs (TAC-2008 corpus). Human evaluation using Amazon Mechanical Turk shows that our system generates the best summary 57.1% of the time. On the other hand, the best answer picked by Yahoo! users is chosen only 31.9% of the time. We also obtain significant higher Pyramid F1 score on the blog task as compared to the system of Lin and Bilmes [2011].
- Within our summarization framework, the statistically learned sentence relevance is included as part of our objective function, whereas previous work on submodular summarization Lin and Bilmes [2011] only uses ngram overlap for query relevance. Additionally, we use Latent Dirichlet Allocation [Blei et al., 2003] to model the topic structure of the sentences, and induce clusterings according to the learned topics. Therefore, our system is capable of generating summaries with broader topic coverage.
- Furthermore, we are the first to study how different metrics for computing text

similarity or dissimilarity affect the quality of submodularity-based summarization methods. Given that submodular summarization highly relies on textual similarity to encourage content coverage and diversity, we investigate measurements based on lexical, semantic, and topical representation. We show empirically that lexical representation-based similarity, such as TFIDF scores, uniformly outperforms semantic similarity computed with WordNet. Moreover, when measuring the summary diversity, topical representation is marginally better than lexical representation, and both of them beats semantic representation.

6.2 Submodular Opinion Summarization

In this section, we describe how query-focused opinion summarization can be addressed by submodular functions combined with dispersion functions. We first define our problem. Then we introduce the components of our objective function (Sections 6.2.1–6.2.3). The full objective function is presented in Section 6.2.4. Lastly, we describe a greedy algorithm with constant factor approximation to the optimal solution for generating summaries (Section 6.2.5).

A set of documents or answers to be summarized are first split into a set of individual sentences $V = \{s_1, \cdots, s_n\}$. Our problem is to select a subset $S \subseteq V$ that maximizes a given objective function $f : 2^V \rightarrow \mathbb{R}$ within a length constraint: $S^* = \arg\max_{S \subseteq V} f(S)$, subject to $| S | \leq c$. $| S |$ is the length of the summary S, and c is the length limit.

Definition 1 *A function $f : 2^V \rightarrow \mathbb{R}$ is submodular iff for all $s \in V$ and every $S \subseteq S' \subseteq V$, it satisfies $f(S \cup \{s\}) - f(S) \geq f(S' \cup \{s\}) - f(S')$.*

Previous submodularity-based summarization work assumes this diminishing return

property makes submodular functions a natural fit for summarization and achieves state-of-the-art results on various datasets. In this paper, we follow the same assumption and work with non-decreasing submodular functions. Nevertheless, they have limitations, one of which is that functions well suited to modeling diversity are not submodular. Recently, Dasgupta et al. [2013] proved that diversity can nonetheless be encoded in well-designed *dispersion functions* which still maintain a constant factor approximation when solved by a greedy algorithm.

Based on these considerations, we propose an objective function $f(S)$ mainly considering three aspects: *relevance* (Section 6.2.1), *coverage* (Section 6.2.2), and *non-redundancy* (Section 6.2.3). Relevance and coverage are encoded in a non-decreasing submodular function, and non-redundancy is enforced by maximizing the dispersion function.

6.2.1 Relevance Function

We first utilize statistical rankers to produce a preference ordering of the candidate answers or sentences. We choose ListNet [Cao et al., 2007], which has been shown to be effective in many information retrieval tasks, as our ranker. We use the implementation from Ranklib [Dang, 2011].

Features used in the ranking algorithm are summarized in Table 6.1. All features are normalized by standardization. Due to the length limit, we cannot provide the full results on feature evaluation. Nevertheless, we find that ranking candidates by TFIDF similarity or key phrases overlapping with the query can produce comparable results with using the full feature set (see Section 6.4).

Basic Features
- answer position in all answers/sentence position in blog - length of the answer/sentence - length is less than 5 words
Sentiment Features
- number/portion of sentiment words from a lexicon (Section 6.2.2) - if contains sentiment words with the same polarity as sentiment words in query
Query-Sentence Overlap Features
- unigram/bigram TF/TFIDF similarity with query - number of key phrases in the query that appear in the sentence. A model similar to that described in Luo et al. [2013] was applied to detect key phrases
Query-Independent Features
- unigram/bigram TFIDF similarity with cluster centroid - sumBasic score [Nenkova and Vanderwende, 2005] - number of topic signature words [Lin and Hovy, 2000] - JS divergence with cluster

Table 6.1: Features used for candidate ranking. We use them for ranking answers in both community QA and blogs.

We take the ranks output by the ranker, and define the relevance of the current summary S as: $r(S) = \sum_i^{|S|} \sqrt{rank_i^{-1}}$, where $rank_i$ is the rank of sentence s_i in V. For QA answer ranking, sentences from the same answer have the same ranking. The function $r(S)$ is our first submodular function.

6.2.2 Coverage Functions

Topic Coverage. This function is designed to capture the idea that a comprehensive opinion summary should provide thoughts on distinct aspects. Topic models such as Latent Dirichlet Allocation (LDA) [Blei et al., 2003] and its variants are able to discover hidden topics or aspects of document collections, and thus afford a natural way to cluster texts according to their topics. Recent work [Xie and Xing, 2013] shows the effectiveness of utilizing topic models for newsgroup document clustering. We first learn an LDA model from the data, and treat each topic as a cluster. We estimate a

sentence-topic distribution $\vec{\theta}$ for each sentence, and assign the sentence to the cluster k corresponding to the mode of the distribution (i.e., $k = \arg\max_i \theta_i$). This naive approach produces comparable clustering performance to the state-of-the-art according to Xie and Xing [2013]. $\mathcal{T}$ is defined as the clustering induced by our algorithm on the set V. The topic coverage of the current summary S is defined as $t(S) = \sum_{T \in \mathcal{T}} \sqrt{|S \cap T|}$. From the concavity of the square root it follows that sets S with uniform coverages of topics are preferred to sets with skewed coverage.

Authorship Coverage. This term encourages the summarization algorithm to select sentences from different authors. Let $\mathcal{A}$ be the clustering induced by the sentence to author relation. In community QA, sentences from the answers given by the same user belong to the same cluster. Similarly, sentences from blogs with the same author are in the same cluster. The authorship score is defined as $a(S) = \sum_{A \in \mathcal{A}} \sqrt{|S \cap A|}$.

Polarity Coverage. The polarity score encourages the selection of summaries that cover both positive and negative opinions. We categorize each sentence simply by counting the number of polarized words given by our lexicon. A sentence belongs to a positive cluster if it has more positive words than negative ones, and vice versa. If any negator co-occurs with a sentiment word (e.g. within a window of size 5), the sentiment is reversed.[2] The polarity clustering $\mathcal{P}$ thus have two clusters corresponding to positive and negative opinions. The score is defined as $p(S) = \sum_{P \in \mathcal{P}} \sqrt{| S \cap P |}$.

Our lexicon consists of MPQA lexicon [Wilson et al., 2005], General Inquirer [Stone et al., 1966], and SentiWordNet [Esuli and Sebastiani, 2006]. Each word in SentiWord-

[2]There exists a large amount of work on determining the polarity of a sentence [Pang and Lee, 2008] which can be employed for polarity clustering in this work. We decide to focus on summarization, and estimate sentence polarity through sentiment word summation [Yu and Hatzivassiloglou, 2003], though we do not distinguish different sentiment words.

Net is associated with a positive score and a negative score, and we only use the words with a polarity score larger than 0.7. Words with conflicting sentiments from different lexicons are removed.

Content Coverage. Similarly to Lin and Bilmes [2011] and Dasgupta et al. [2013], we use the following function to measure content coverage of the current summary S: $c(S) = \sum_{v \in V} \min(cov(v, S), \theta \cdot cov(v, V))$, where $cov(v, S) = \sum_{u \in S} sim(v, u)$. $sim(v, u)$ measures the similarity between sentences v and u. This function is utilized to make sure the summary will not over-concentrate on a normalsize portion of the document which may lead to poor coverage of the documents. We set θ to 0.25 according to Dasgupta et al. [2013].

We experiment with two types of similarity functions. One is a Cosine TFIDF similarity score. The other is a WordNet-based semantic similarity score between pairwise dependency relations from two sentences [Dasgupta et al., 2013]. Specifically, $sim_{Sem}(v, u) = \sum_{rel_i \in v, rel_j \in u} WN(a_i, a_j) \times WN(b_i, b_j)$, where $rel_i = (a_i, b_i), rel_j = (a_j, b_j)$, $WN(w_i, w_j)$ is the shortest path length between words w_i and w_j in WordNet graph. All scores are scaled onto $[0, 1]$.

6.2.3 Dispersion Function

Summaries should contain as little redundant information as possible. We achieve this by adding an additional term to the objective function, encoded by a dispersion function. Given a set of sentences S, a complete graph is constructed with each sentence in S as a node. The weight of each edge (u, v) is their dissimilarity $d'(u, v)$. Then the distance between any pair of u and v, $d(u, v)$, is defined as the total weight of the shortest path

connecting u and v.[3] We experiment with two forms of dispersion function [Dasgupta et al., 2013]: (1) $h_{sum} = \sum_{u,v \in V, u \neq v} d(u, v)$, and (2) $h_{min} = \min_{u,v \in V, u \neq v} d(u, v)$.

Then we need to define the dissimilarity function $d'(\cdot, \cdot)$. There are different ways to measure the dissimilarity between sentences [Mihalcea et al., 2006, Agirre et al., 2012]. In this work, we experiment with three types of dissimilarity functions.

Lexical Dissimilarity. This function is based on the well-known Cosine similarity score using TFIDF weights. Let $sim_{tfidf}(u, v)$ be the Cosine similarity between u and v, then we have $d'_{Lex}(u, v) = 1 - sim_{tfidf}(u, v)$.

Semantic Dissimilarity. This function is based on the semantic meaning embedded in the dependency relations. $d'_{Sem}(u, v) = 1 - sim_{Sem}(v, u)$, where $sim_{Sem}(v, u)$ is the semantic similarity used in content coverage measurement in Section 6.2.2.

Topical Dissimilarity. We propose a novel dissimilarity measure based on topic models. Celikyilmaz et al. [2010] show that estimating the similarity between query and passages by using topic structures can help improve the retrieval performance. As discussed in the topic coverage in Section 6.2.2, each sentence is represented by its sentence-topic distributions estimated by LDA. For candidate sentence u and v, let their topic distributions be P_u and P_v. Then the dissimilarity between u and v can be defined as: $d'_{Topic}(u, v) = JSD(P_u || P_v) = \frac{1}{2}(\sum_i P_u(i) \log_2 \frac{P_u(i)}{P_a(i)} + \sum_i P_v(i) \log_2 \frac{P_v(i)}{P_a(i)})$ where $P_a(i) = \frac{1}{2}(P_u(i) + P_v(i))$.

[3] This definition of distance is used to produce theoretical guarantees for the greedy algorithm described in Section 6.2.5.

6.2.4 Full Objective Function

The objective function takes the interpolation of the submodular functions and dispersion function:

$$\mathcal{F}(S) = r(S) + \alpha t(S) + \beta a(S) + \gamma p(S) + \eta c(S) + \delta h(S). \tag{6.1}$$

The coefficients $\alpha, \beta, \gamma, \eta, \delta$ are non-negative real numbers and can be tuned on a development set.[4] Notice that each summand except $h(S)$ is a non-decreasing, non-negative, and submodular function, and summation preserves monotonicity, non-negativity, and submodularity. Dispersion function $h(s)$ is either h_{sum} or h_{min} as introduced previously.

6.2.5 Summary Generation via Greedy Algorithm

Generating the summary that maximizes our objective function in Equation 6.1 is NP-hard [Chandra and Halldórsson, 1996]. We choose to use a greedy algorithm that guarantees to obtain a constant factor approximation to the optimal solution [Nemhauser et al., 1978, Dasgupta et al., 2013]. Concretely, starting with an empty set, for each iteration, we add a new sentence so that the current summary achieves the maximum value of the objective function. In addition to the theoretical guarantee, existing work [McDonald, 2007] has empirically shown that classical greedy algorithms usually works near-optimally.

[4]The values for the coefficients are 5.0, 1.0, 10.0, 5.0, 10.0 for $\alpha, \beta, \gamma, \eta, \delta$, respectively, as tuned on the development set.

6.3 Experimental Setup

6.3.1 Opinion Question Identification

We first build a classifier to automatically detect opinion oriented questions in Community QA; questions in the blog dataset are all opinionated. Our opinion question classifier is trained on two opinion question datasets: (1) the first, from Li et al. [2008a], contains 646 opinionated and 332 objective questions; (2) the second dataset, from Amiri et al. [2013], consists of 317 implicit opinion questions, such as "*What can you do to help environment?*", and 317 objective questions. We train a RBF kernel based SVM classifier to identify opinion questions, which achieves F1 scores of 0.79 and 0.80 on the two datasets when evaluated using 10-fold cross-validation (the best F1 scores reported are 0.75 and 0.79).

6.3.2 Datasets

Community QA Summarization: Yahoo! Answers. We use the Yahoo! Answers dataset from Yahoo! *Webscope*^{TM} program,[5] which contains 3,895,407 questions. We first run the opinion question classifier to identify the opinion questions. For summarization purpose, we require each question having at least 5 answers, with the average length of answers larger than 20 words. This results in 130,609 questions.

To make a compelling task, we reserve questions with an average length of answers larger than 50 words as our test set for both ranking and summarization; all the other questions are used for training. As a result, we have 92,109 questions in the training set

[5] `http://sandbox.yahoo.com/`

for learning the statistical ranker, and 38,500 in the test set. The category distribution of training and test questions (Yahoo! Answers organizes the questions into predefined categories) are similar. 10,000 questions from the training set are further reserved as the development set. Each question in the Yahoo! Answers dataset has a user-voted best answer. These best answers are used to train the statistical ranker that predicts relevance. Separate topic models are learned for each category, where the category tag is provided by Yahoo! Answer.

Blog Summarization: TAC 2008. We use the TAC 2008 corpus [Dang, 2008], which consists of 25 topics. 23 of them are provided with human labeled nuggets, which TAC used in human evaluation. TAC also provides snippets (i.e., sentences) that are frequently retrieved by participant systems or identified as relevant by human annotators. We do not assume those snippets are known to any of our systems.

6.3.3 Comparisons

For both opinion summarization tasks, we compare with (1) the approach by Dasgupta et al. [2013], and (2) the systems from Lin and Bilmes [2011] with and without query information. The sentence clustering process in Lin and Bilmes [2011] is done by using CLUTO [Karypis, 2003]. For the implementation of systems in Lin and Bilmes [2011] and Dasgupta et al. [2013], we always use the parameters reported to have the best performance in their work.

For cQA summarization, we use the **best answer** voted by the user as a baseline. Note that this is a strong baseline since all the other systems are unaware of which answer is the best. For blog summarization, we have three additional baselines – the

best systems in TAC 2008 [Kim et al., 2008, Li et al., 2008b], top sentences returned by our **ranker**, a baseline produced by TFIDF similarity and a lexicon (henceforth called **TFIDF+Lexicon**). In TFIDF+Lexicon, sentences are ranked by the TFIDF similarity with the query, and then sentences with sentiment words are selected in sequence. This baseline aims to show the performance when we only have access to lexicons without using a learning algorithm.

6.4 Results

6.4.1 Evaluating the Ranker

We evaluate our ranker (described in Section 6.2.1) on the task of best answer prediction. Table 6.2 compares the average precision and mean reciprocal rank (MRR) of our method to those of three baselines, (1) where answers are ranked randomly (**Baseline (Random)**), (2) by length (**Baseline (Length)**), and (3) by Jensen Shannon Divergence (**JSD**) with all answers. We expect that the best answer is the one that covers the most information, which is likely to have a normalsizeer JSD. Therefore, we use JSD to rank answers in the ascending order. Table 6.2 manifests that our ranker outperforms all the other methods.

	Baseline (Random)	**Baseline (Length)**	**JSD**	**Ranker (ListNet)**
Avg Precision	0.1305	0.2834	0.4000	**0.5336**
MRR	0.3403	0.4889	0.5909	**0.6496**

Table 6.2: Performance for best answer prediction. Our ranker outperforms the three baselines.

6.4.2 Community QA Summarization

Automatic Evaluation. Since human written abstracts are not available for the Yahoo! Answers dataset, we adopt the Jensen-Shannon divergence (JSD) to measure the summary quality. Intuitively, a normalsizeer JSD implies that the summary covers more of the content in the answer set. Louis and Nenkova [2013] report that JSD has a strong negative correlation (Spearman correlation = -0.737) with the overall summary quality for multi-document summarization (MDS) on news articles and blogs. Our task is similar to MDS. Meanwhile, the average JSD of the best answers in our test set is normalsizeer than that of the other answers (0.39 vs. 0.49), with an average length of 103 words compared with 67 words for the other answers. Also, on the blog task (Section 6.4.3), the top two systems by JSD also have the top two ROUGE scores (a common metric for summarization evaluation when human-constructed summaries are available). Thus, we conjecture that JSD is a good metric for community QA summaries.

Table 6.3 shows that our system using a content coverage function based on Cosine using TFIDF weights, and a dispersion function (h_{sum}) based on lexicon dissimilarity and 100 topics, outperforms all of the compared approaches (paired-t test, $p < 0.05$). The topic number is tuned on the development set, and we find that varying the number of topics does not impact performance too much. Meanwhile, both our system and Dasgupta et al. [2013] produce better JSD scores than the two variants of the Lin and Bilmes [2011] system, which implies the effectiveness of the dispersion function. We further examine the effectiveness of each component that contributes to the objective function (Section 6.2.4), and the results are shown in Table 6.4.

	Length	
	100	200
Best answer	0.3858	-
Lin and Bilmes [2011]	0.3398	0.2008
Lin and Bilmes [2011] + q	0.3379	0.1988
Dasgupta et al. [2013]	0.3316	0.1939
Our system	**0.3017**	**0.1758**

Table 6.3: Summaries evaluated by Jensen-Shannon divergence (JSD) on Yahoo Answer for summaries of 100 words and 200 words. The average length of the best answer is 102.70.

	$\mathbf{JSD}_{100}$	$\mathbf{JSD}_{200}$
Rel(evance)	0.3424	0.2053
Rel + Aut(hor)	0.3375	0.2040
Rel + Aut + TM (Topic Models)	0.3366	0.2033
Rel + Aut + TM + Pol(arity)	0.3309	0.1983
Rel + Aut + TM + Pol + Cont(ent Coverage)	0.3102	0.1851
Rel + Aut + TM + Pol + Cont + Disp(ersion)	**0.3017**	**0.1758**

Table 6.4: Value addition of each component in the objective function. The JSD on each line is statistically significantly lower than the JSD on the previous ($\alpha = 0.05$).

Human Evaluation. Human evaluation for Yahoo! Answers is carried out on Amazon Mechanical Turk[6] with carefully designed tasks (or "HITs"). Turkers are presented summaries from different systems in a random order, and asked to provide two rankings, one for overall quality and the other for information diversity. We indicate that informativeness and non-redundancy are desirable for quality; however, Turkers are allowed to consider other desiderata, such as coherence or responsiveness, and write down those when they submit the answers. Here we believe that ranking the summaries is easier than evaluating each summary in isolation Lerman et al. [2009]. Turkers were optionally asked to provide a brief comment on their rankings.

We randomly select 100 questions from our test set, each of which is evaluated by 4 distinct Turkers located in United States. 40 HITs are thus created, each containing 10

[6] `https://www.mturk.com/mturk/`

different questions. Four system summaries (best answer, Dasgupta et al. [2013], and our system with 100 and 200 words respectively) are displayed along with one noisy summary (i.e. irrelevant to the question) per question in random order.[7] We reject Turkers' HITs if they rank the noisy summary higher than any other. Two duplicate questions are added to test intra-annotator agreement. We reject HITs if Turkers produced inconsistent rankings for both duplicate questions. A total of 137 submissions of which 40 HITs pass the above quality filters.

Turkers of all accepted submissions report themselves as native English speakers. An inter-rater agreement of Fleiss' κ of 0.28 (fair agreement [Landis and Koch, 1977]) is computed for quality ranking and κ is 0.43 (moderate agreement) for diversity ranking. Table 6.5 shows the percentage of times a particular method is picked as the best summary, and the macro-/micro-average rank of a method, for both overall quality and information diversity. Macro-average is computed by first averaging the ranks per question and then averaging across all questions.

	Len. of Summary	**Overall Quality**			**Information Diversity**		
		%	Average Rank		%	Average Rank	
		Best	Macro	Micro	Best	Macro	Micro
Best answer	102.70	31.9%	2.68	2.69	9.6%	3.27	3.29
Dasgupta et al. [2013]	100	11.0%	2.84	2.83	5.0%	2.95	2.94
Our system		12.5%	**2.50***	**2.50***	6.7%	**2.43***	**2.43***
Our system	200	**44.6%**	**1.98***	**1.98***	**78.7%**	**1.35***	**1.34***

Table 6.5: Human evaluation on Yahoo! Answer Data. **Boldface** implies statistically significance compared to other results in the same columns using paired-t test. Both of our systems are ranked higher (i.e. numbers in **bold** with *) than the best answers voted by Yahoo! users and system summaries from Dasgupta et al. [2013].

For overall quality, our system with a 200 word limit is selected as the best in 44.6% of the evaluations. It outperforms the best answer (31.9%) significantly, which suggests that our system summary covers relevant information that is not contained in the best

[7]Note that we aim to compare results with the gold-standard best answers of about 100 words. The evaluation of the 200-word summaries is provided only as an additional data-point.

answer. Our system with a length constraint of 100 words is chosen as the best for quality 12.5% times while that of Dasgupta et al. [2013] is chosen 11.0% of the time. Our system is also voted as the best summary for diversity in 78.7% of the evaluations. More interestingly, both of our systems, with 100 words and 200 words, outperform the best answer and Dasgupta et al. [2013] for average ranking (both overall quality and information diversity) significantly by using Wilcoxon signed-rank test ($p < 0.05$). When we check the reasons given by Turkers, we found that people usually prefer our summaries due to "helpful suggestions that covered many options" or being "balanced with different opinions". When Turks prefer the best answers, they mostly stress on coherence and responsiveness. Sample summaries from all the systems are displayed in Figure 6.2.

6.4.3 Blog Summarization

Automatic Evaluation. We use the ROUGE [Lin and Hovy, 2003] software with standard options to automatically evaluate summaries with reference to the human labeled nuggets as those are available for this task. ROUGE-2 measures bigram overlap and ROUGE-SU4 measures the overlap of unigram and skip-bigram separated by up to four words. We use the ranker trained on Yahoo! data to produce relevance ordering, and adopt the system parameters from Section 6.4.2. Table 6.6 shows that our system outperforms the best system in TAC'08 with highest ROUGE-2 score [Kim et al., 2008], the two baselines (TFIDF+Lexicon, and our ranker), Lin and Bilmes [2011], and Dasgupta et al. [2013].

Human Evaluation. For human evaluation, we use the standard Pyramid F-score used in the TAC'08 opinion summarization track with $\beta = 3$. According to Dang [2008] that

Question: What is the long term effect of piracy on the music and film industry?

Dasgupta et al. [2013] (Qty Rank=2.75 Div. Rank=2.5):

•In short, I don't think the music industry in particular will ever enjoy the huge profits of the 90's.

•Please-People in those businesses make millions of dollars as it is !! I don't think piracy hurts them at all !!!

•The other thing will be music and movies with less quality.

•Its a big gray area, I dont see anything wrong with burning a mix cd or a cd for a friend so long as youre not selling them for profit.

•By removing the profitability of music & film companies, piracy takes away their motivation to produce new music & movies.

Our system (100 words) (Qty Rank=2.25 Div. Rank=2.25):

•Rising costs for movies and music. The other thing will be music and movies with less quality.

•Now, with piracy, there isn't the willingness to take chances.

•But it's also like the person put the effort into it and they aren't getting paid. It's a big gray area, I don't see anything wrong with burning a mix cd or a cd for a friend so long as you're not selling them for profit.

•It is forcing them to rework their business model, which is a good thing.

Our system (200 words) (Qty. Rank=2.25, Div Rank=1.25):

•Rising costs for movies and music. The other thing will be music and movies with less quality.

•Now, with piracy, there isn't the willingness to take chances. American Idol is the result of this. .... The real problem here is that the mainstream music will become even tighter. Record labels will not won't to go far from what is currently like by the majority.

•I hate when people who have billions of dollars whine about not having more money. But it's also like the person put the effort into it and they aren't getting paid ... I don't see anything wrong with burning a mix cd or a cd for a friend ....

•It is forcing them to rework their business model, which is a good thing.

•By removing the profitability of music & film companies, piracy takes away their motivation to produce new music & movies.

Figure 6.2: Sample summaries from Dasgupta et al. [2013], and our systems (100 words and 200 words). Sentences from separate bullets (•) are partial answers from different users.

overall responsiveness of the summaries highly correlates with pyramid scores when β is between 3 and 5. In the TAC task, systems are allowed to return up to 7,000 non-white characters for each question. Since the TAC metric favors recall we do not produce

	ROUGE-2	ROUGE-SU4	JSD
Best system in TAC'08	0.2923	0.3766	0.3286
TFIDF + Lexicon	0.3069	0.3876	0.2429
Ranker (ListNet)	0.3200	0.3960	0.2293
Lin and Bilmes [2011]	0.2732	0.3582	0.2330
Lin and Bilmes [2011] + q	0.2852	0.3700	0.2349
Dasgupta et al. [2013]	0.2618	0.3500	0.2370
Our system	**0.3234**	**0.3978**	*0.2258*

Table 6.6: Results on TAC'08 dataset. Our system has significant better ROUGE scores than all the other systems except our ranker (paired-t test, $p < 0.05$). We also achieve the best JS divergence.

	Pyramid F-score
Best system in TAC'08	0.2225
Lin and Bilmes [2011]	0.2790
Our system	**0.3620**

Table 6.7: Human evaluation with Pyramid F-score. Our system significantly outperforms the others.

summaries shorter than 7,000 characters. We ask two human judges to evaluate our system along with the one that got the highest Pyramid F-score in the TAC'08 and Lin and Bilmes [2011]. The results are displayed in Table 6.7. Cohen's κ for inter-annotator agreement is 0.68 (substantial). While we did not explicitly evaluate non-redundancy, both of our judges report that our system summaries contain less redundant information.

6.4.4 Further Discussion

Given that the text similarity metrics and dispersion functions play important roles in the framework, we further study the effectiveness of different content coverage functions (Cosine using TFIDF vs. Semantic), dispersion functions (h_{sum} vs. h_{min}), and dissimilarity metrics used in dispersion functions (Semantic vs. Topical vs. Lexical). Results on Yahoo! Answer (Table 6.8 show that systems using summation of distances for dis-

Yahoo! Answer				
	DISPERSION$_{sum}$		DISPERSION$_{min}$	
DISSIMI	Cont$_{tfidf}$	Cont$_{sem}$	Cont$_{tfidf}$	Cont$_{sem}$
Semantic	0.3143	0.324 3	0.3129	0.3232
Topical	0.3101	0.3202	0.3106	0.3209
Lexical	**0.3017**	0.3147	0.3071	0.3172

Table 6.8: Effect of different dispersion functions, content coverage, and dissimilarity metrics on our system. JSD values for different combinations on Yahoo! data, using LDA with 100 topics. All systems are significantly different from each other at significance level $\alpha = 0.05$. Systems using summation of distances for dispersion function (h_{sum}) uniformly outperform the ones using minimum distance (h_{min}).

persion functions (h_{sum}) uniformly outperform the ones using minimum distance (h_{min}). Meanwhile, Cosine using TFIDF is better at measuring content coverage than WordNet-based semantic measurement, and this may due to the limited coverage of WordNet on verbs. This is also true for dissimilarity metrics, where lexical dissimilarity outperforms topical and semantic measures.

TAC 2008				
	DISPERSION$_{sum}$		DISPERSION$_{min}$	
DISSIMI	Cont$_{tfidf}$	Cont$_{sem}$	Cont$_{tfidf}$	Cont$_{sem}$
Semantic	0.2216	0.2169	0.2772	0.2579
Topical	0.2128	0.2090	**0.3234**	0.3056
Lexical	0.2167	0.2129	0.3117	0.3160

Table 6.9: Effect of different dispersion functions, content coverage, and dissimilarity metrics on our system. ROUGE scores of different choices for TAC 2008 data. All systems use LDA with 40 topics. The parameters of our systems are adopted from the ones tuned on Yahoo! Answers.

Results on blog data (Table 6.9, however, show that using minimum distance for dispersion produces better results. This indicates that optimal dispersion function varies by genre. Topical-based dissimilarity also marginally outperforms the other two metrics in blog data. This further corroborates the conclusion of Dasgupta et al. [2013] that the optimal dispersion function varies by genre.

6.5 Conclusion

In this chapter, we proposed a submodular function-based opinion summarization framework. Tested on community QA and blog summarization, our approach outperformed state-of-the-art methods that are also based on submodularity in both automatic evaluation and human evaluation. Our framework is capable of including statistically learned sentence relevance and encouraging the summary to cover diverse topics. We also studied different metrics for text similarity estimation and their effect on summarization.

CHAPTER 7

SOCIALLY-INFORMED TIMELINE GENERATION

In this chapter, we will describe a socially-informed timeline generation system, which utilizes information from both news articles and user comments to enrich users' reading experience.

7.1 Introduction

Social media sites on the Internet provide increasingly more, and increasingly popular, means for people to voice their opinions on trending events. Traditional news media — the New York Times and CNN, for example — now provide online mechanisms that allow and encourage readers to share reactions, opinions, and personal experiences relevant to a news story. For complex emerging events, in particular, user comments can provide relevant, interesting and insightful information beyond the facts reported in the news. But their large volume and tremendous variation in quality make it impossible for readers to efficiently digest the user-generated content, much less integrate it with reported facts from the dozens or hundreds of news reports produced on the event each day.

In this work, we present a *socially-informed timeline generation system* that jointly generates a news article summary and a user comment summary for each day of an ongoing complex event. A sample (gold standard) timeline snippet for Ukraine Crisis is shown in Figure 7.1. The event timeline is on the left; the comment summary for March 17^{th} is on the right.

While generating timelines from news articles and summarizing user comments have

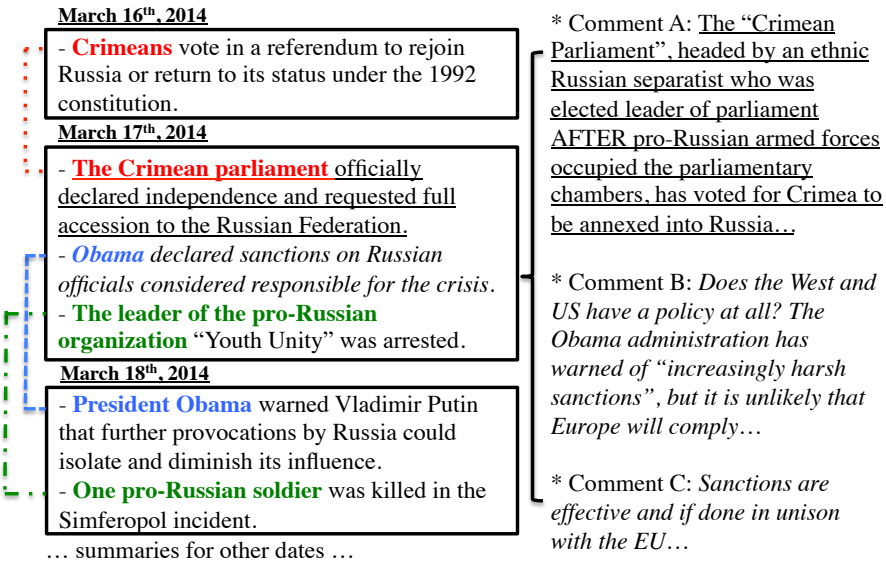


Figure 7.1: A snippet of the event timeline on Ukraine Crisis is displayed on the left. On the right, we display a set of representative comments addressing the article summary of March 17^{th}. Comment A (underlined) brings a perspective on "Crimean parliament passes declaration of independence" (the article sentence is also underlined on the left). Comments B and C focus on Obama's sanctions on Ukrainian and Russian officials. Sentences linked by edges belong to the same event thread, which is centered on the entities with the same color.

been studied as separate problems [Yan et al., 2011, Ma et al., 2012], their joint summarization for timeline generation raises new challenges. Firstly, there should be a tight connection between the article and comment portion of the timeline. By definition, users comment on socially relevant events. So the important part of articles and insightful comments should both cover these events. Moreover, good reading experience requires that the article summary and comment summary demonstrate evident connectivity. For example, Comment C in Figure 7.1 ("Sanctions are effective and if done in unison with the EU") is obscure without knowing the context that "sanctions are imposed by U.S". Simply combining the outputs from a timeline generation system and a comment summarization system may lead to timelines that lack cohesion. On the other hand, articles and comments are from intrinsically different genres of text: articles emphasize facts and are written in a professional style; comments reflect opinions in a less formal way.

Thus, it could be difficult to recognize the connections between articles and comments. Finally, it is also challenging to enforce continuity in timelines with many entities and events.

To address the challenges mentioned above, we formulate the timeline generation task as an optimization problem, where we maximize topic cohesion between the article and comment summaries while preserving their ability to reflect *important* concepts and subevents, adequate *coverage* of mentioned topics, and *continuity* of the timeline as it is updated with new material each day. We design a novel alternating optimizing algorithm that allows the generation of a high quality article summary and comment summary via mutual reinforcement. We demonstrate the effectiveness of our algorithm on four disparate complex event datasets collected over months from the New York Times, CNN, and BBC. Automatic evaluation using ROUGE [Lin and Hovy, 2003] and gold standard timelines indicates that our system can effectively leverage user comments to outperform state-of-the-art approaches on timeline generation. In a human evaluation via Amazon Mechanical Turk, the comment summaries generated by our method were selected as the best in terms of informativeness and insightfulness in 66.7% and 51.7% of the evaluations (vs. 26.7% and 30.0% for randomly selected editor's-picks).

Especially, our optimization framework relies on two scoring functions that estimate the importance of including individual article sentences and user comments in the timeline. Based on the observation that entities or events frequently discussed in the user comments can help with identify summary-worthy content, we show that the scoring functions can be learned jointly by utilizing graph-based regularization. Experiments show that our joint learning model outperforms state-of-the-art ranking algorithms and other joint learning based methods when evaluated on sentence ranking and comment ranking. For example, we achieve an NDCG@3 of 0.88 on the Ukraine crisis dataset,

compared to 0.77 from Yang et al. [2011] which also conducts joint learning between articles and social context using factor graphs.

Finally, to encourage continuity in the generated timeline, we propose an entity-centered event threading algorithm. Human evaluation demonstrates that users who read timelines with event threads write more informative answers than users who do not see the threads while answering the same questions. This implies that our system constructed threads can help users better navigate the timelines and collect relevant information in a short time.

7.2 Data Collection and Preprocessing

We crawled news articles from New York Times (NYT), CNN, and BBC on four trending events: the missing Malaysia Airlines Flight MH370 (MH370), the political unrest in Ukraine (Ukraine), the Israel-Gaza conflict (Israel-Gaza), and the NSA surveillance leaks (NSA). For each event, we select a set of key words (usually entities' name), which are used to filter out irrelevant articles. We collect comments for NYT articles through NYT community API, and comments for CNN articles via Disqus API. [1] NYT comments come with information on whether a comment is an editor's-pick. The statistics on the four datasets are displayed in Table 7.1.[2]

	Time Span	**# Articles**	**# Comments**
MH370	03/08 - 06/30	955	406,646
Ukraine	03/08 - 06/30	3,779	646,961
Israel-Gaza	07/20 - 09/30	909	322,244
NSA	03/23 - 06/30	145	60,481

Table 7.1: Statistics on the four event datasets.

[1] BBC comment volume is low, so we do not collect it.

[2] The datasets are available at `http://www.cs.cornell.edu/~luwang/data.html`.

We extract parse trees, dependency trees, and coreference resolution results of articles and comments with Stanford CoreNLP [Manning et al., 2014]. Sentences in articles are labeled with timestamps using SUTime [Chang and Manning, 2012].

We also collect all articles with comments from NYT in 2013 (henceforth NYT2013) to form a training set for learning importance scoring functions on articles sentences and comments (see Section 7.3). NYT2013 contains 3,863 articles and 833,032 comments.

7.3 Joint Learning for Importance Scoring

We first introduce a joint learning method that uses *graph-based regularization* to simultaneously learn two functions — a SENTENCE scorer and a COMMENT scorer — that predict the importance of including an individual news article sentence or a particular user comment in the timeline.

We train the model on the aforementioned NYT2013 dataset, where 20% of the articles and their comments are reserved for parameter tuning. Formally, the training data consists of a set of articles $D = \{d_i\}_{i=0}^{|D|-1}$. Each article d_i contains a set of sentences $x_{s_{d_i}} = \{x_{s_{d_i},j}\}_{j=0}^{|s_{d_i}|-1}$ and a set of associated comments $x_{c_{d_i}} = \{x_{c_{d_i},k}\}_{k=0}^{|c_{d_i}|-1}$, where $|s_{d_i}|$ and $|c_{d_i}|$ are the numbers of sentences and comments for d_i. For simplicity, we use x_s or x_c to denote a sentence or a comment wherever there is no ambiguity.

In addition, each article has a human-written abstract. We use the ROUGE-2 [Lin and Hovy, 2003] score of each sentence computed against the associated abstract as its gold-standard importance score. Each comment is assigned a gold-standard value of 1.0 if it is an editor's pick, or 0.0 otherwise.

The SENTENCE and COMMENT scorers rely on two classifiers, each designed to

handle the special characteristics of news and user comments, respectively; and a graph-based regularizing constraint that encourages similarity between selected sentences and comments. We describe each component below.

Article SENTENCE **Importance.** Each sentence x_s in a news article is represented as a k-dimensional feature vector $\mathbf{x_s} \in \mathbb{R}^k$, with a gold-standard label y_s. We denote the training set as a feature matrix $\tilde{\mathbf{X}}_\mathbf{s}$, with a label vector $\tilde{\mathbf{Y}}_\mathbf{s}$. To produce the SENTENCE scoring function $f_s(x_s) = \mathbf{x_s} \cdot \mathbf{w_s}$, we use ridge regression to learn a vector $\mathbf{w_s}$ that minimizes $||\tilde{\mathbf{X}}_\mathbf{s}\mathbf{w_s} - \tilde{\mathbf{Y}}_\mathbf{s}||_2^2 + \beta_s \cdot ||\mathbf{w_s}||_2^2$. Features used in the model are listed in Table 7.2.

We also impose the following *position-based regularizing constraint* to encode the fact that the first sentence in a news article usually conveys the most essential information: $\lambda_s \cdot \sum_{d_i} \sum_{x_{s_{d_i}, j}, j \neq 0} ||(\mathbf{x_{s_{d_i},0}} - \mathbf{x_{s_{d_i},j}}) \cdot \mathbf{w_s} - (y_{s_{d_i},0} - y_{s_{d_i},j})||_2^2$, where $x_{s_{d_i},j}$ is the j-th sentence in document d_i. Term $(\mathbf{x_{s_{d_i},0}} - \mathbf{x_{s_{d_i},j}}) \cdot \mathbf{w_s}$ measures the difference in predicted scores between the first sentence and any other sentence. This value is expected be close to the true difference. We further construct $\tilde{\mathbf{X}}'_\mathbf{s}$ to contain all difference vectors $(\mathbf{x_{s_{d_i},0}} - \mathbf{x_{s_{d_i},j}})$, with $\tilde{\mathbf{Y}}'_\mathbf{s}$ as label difference vector. The objective function to minimize becomes

$$J_s(\mathbf{w_s}) = ||\tilde{\mathbf{X}}_\mathbf{s}\mathbf{w_s} - \tilde{\mathbf{Y}}_\mathbf{s}||_2^2 + \lambda_s \cdot ||\tilde{\mathbf{X}}'_\mathbf{s}\mathbf{w_s} - \tilde{\mathbf{Y}}'_\mathbf{s}||_2^2 + \beta_s \cdot ||\mathbf{w_s}||_2^2 \tag{7.1}$$

Basic Features	**Social Features**
- number of words	- avg/sum frequency of words appearing in comment
- absolute/relative position	- avg/sum frequency of dependency relations
- overlaps with headline	appearing in comment
- avg/sum TF-IDF scores	
- number of NEs	

Table 7.2: Features used for sentence importance scoring.

User COMMENT **Importance.** Similarly, each comment x_c is represented as an $l-$dimensional feature vector $\mathbf{x_c} \in \mathbb{R}^l$, with label y_c. Comments in the training data

are denoted with a feature matrix $\tilde{\mathbf{X}}_\mathbf{c}$ with a label vector $\tilde{\mathbf{Y}}_\mathbf{c}$. Likewise, we learn $f_c(x_c) = \mathbf{x_c} \cdot \mathbf{w_c}$ by minimizing $||\tilde{\mathbf{X}}_\mathbf{c}\mathbf{w_c} - \tilde{\mathbf{Y}}_\mathbf{c}||_2^2 + \beta_c \cdot ||\mathbf{w_c}||_2^2$. Features are listed in Table 7.3.

We apply a *pairwise preference-based regularizing constraint* [Joachims, 2002] to incorporate a bias toward editor's picks: $\lambda_c \cdot \sum_{d_i} \sum_{x_{c_{d_i},j} \in \mathbf{E_{d_i}}, x_{c_{d_i},k} \notin \mathbf{E_{d_i}}} ||(\mathbf{x_{c_{d_i},j}} - \mathbf{x_{c_{d_i},k}}) \cdot \mathbf{w_c} - 1||_2^2$, where $\mathbf{E_{d_i}}$ are the editor's picks for d_i. Term $(\mathbf{x_{c_{d_i},j}} - \mathbf{x_{c_{d_i},k}}) \cdot \mathbf{w_c}$ enforces the separation of editor's picks from regular comments. We further construct $\tilde{\mathbf{X}}'_\mathbf{c}$ to contain all the pairwise differences $(\mathbf{x_{c_{d_i},j}} - \mathbf{x_{c_{d_i},k}})$. $\tilde{\mathbf{Y}}'_\mathbf{c}$ is a vector of same size as $\tilde{\mathbf{X}}'_\mathbf{c}$ with each element as 1. Thus, the objective function to minimize is:

$$J_c(\mathbf{w_c}) = ||\tilde{\mathbf{X}}_\mathbf{c}\mathbf{w_c} - \tilde{\mathbf{Y}}_\mathbf{c}||_2^2 + \lambda_c \cdot ||\tilde{\mathbf{X}}'_\mathbf{c}\mathbf{w_c} - \tilde{\mathbf{Y}}'_\mathbf{c}||_2^2 + \beta_c \cdot ||\mathbf{w_c}||_2^2 \tag{7.2}$$

Graph-Based Regularization. The regularizing constraint is based on two mutually reinforcing hypotheses: (1) the importance of a sentence depends partially on the availability of sufficient insightful comments that touch on topics in the sentence; (2) the importance of a comment depends partially on whether it addresses notable events reported in the sentences. For example, we want our model to bias $\mathbf{w_s}$ to predict a high score for a sentence with high similarity to numerous insightful comments.

We first create a bipartite graph from sentences and comments on the same articles, where edge *weights* are based on the content similarity between a sentence and a comment (TF-IDF similarity is used). Let $\tilde{\mathbf{R}}$ be an $N \times M$ adjacency matrix, where N and M are the numbers of sentences and comments. R_{sc} is the similarity between sentence x_s and comment x_c. We normalize $\tilde{\mathbf{R}}$ by $\tilde{\mathbf{Q}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{R}}\tilde{\mathbf{D}'}^{-\frac{1}{2}}$, where $\tilde{\mathbf{D}}$ and $\tilde{\mathbf{D}}'$ are diagonal matrices: $\tilde{\mathbf{D}} \in \mathbb{R}^{N \times N}$, $D_{i,i} = \sum_{j=1}^{M} R_{i,j}$; $\tilde{\mathbf{D}}' \in \mathbb{R}^{M \times M}$, $D'_{j,j} = \sum_{i=1}^{N} R_{i,j}$. The interplay between

Basic Features
- number of words
- number of sentences
- avg number of words per sentence
- number of NEs
- number/proportion of capitalized words
- avg/sum TF-IDF
- contains URL
- user rating (pos/neg)
Readability Features
- Flesch-Kincaid Readability
- Gunning-Fog Readability
Discourse Features
- number/proportion of connectives
- number/proportion of hedge words
Article Features
- TF/TF-IDF similarity with article
- TF/TF-IDF similarity with comments
- JS/KL divergence (div) with article
- JS/KL div with comments
Sentiment Features
- number/proportion of positive/negative/neutral words (MPQA [**?**], General Inquirer [Stone et al., 1966])
- number/proportion of sentiment words

Table 7.3: Features used for comment importance scoring.

the two types of data is encoded in the following regularizing constraint:

$$J_{s,c}(\mathbf{w_s}, \mathbf{w_c}) = \lambda_{sc} \cdot \sum_{d_i} \sum_{x_s \in x_{s_{d_i}}, x_c \in x_{c_{d_i}}} Q_{x_s,x_c} \cdot (\mathbf{x_s} \cdot \mathbf{w_s} - \mathbf{x_c} \cdot \mathbf{w_c})^2 \tag{7.3}$$

Full Objective Function. Thus, the full objective function consists of the three parts discussed above:

$$J(\mathbf{w_s}, \mathbf{w_c}) = J_s(\mathbf{w_s}) + J_c(\mathbf{w_c}) + J_{s,c}(\mathbf{w_s}, \mathbf{w_c}) \tag{7.4}$$

Furthermore, using the following notation,

$$\tilde{\mathbf{X}} = \begin{bmatrix} \tilde{\mathbf{X}}_\mathbf{s} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{X}}_\mathbf{c} \end{bmatrix} \quad \tilde{\mathbf{Y}} = \begin{bmatrix} \tilde{\mathbf{Y}}_\mathbf{s} \\ \tilde{\mathbf{Y}}_\mathbf{c} \end{bmatrix} \quad \tilde{\mathbf{X}}' = \begin{bmatrix} \tilde{\mathbf{X}}'_\mathbf{s} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{X}}'_\mathbf{c} \end{bmatrix} \quad \tilde{\mathbf{Y}}' = \begin{bmatrix} \tilde{\mathbf{Y}}'_\mathbf{s} \\ \tilde{\mathbf{Y}}'_\mathbf{c} \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} \mathbf{w_s} \\ \mathbf{w_c} \end{bmatrix}$$

$$\tilde{\boldsymbol{\beta}} = \begin{bmatrix} \beta_s \mathbf{I_k} & \mathbf{0} \\ \mathbf{0} & \beta_c \mathbf{I_l} \end{bmatrix} \quad \tilde{\boldsymbol{\lambda}} = \begin{bmatrix} \lambda_s \mathbf{I}_{|\mathbf{X_s'}|} & \mathbf{0} \\ \mathbf{0} & \lambda_c \mathbf{I}_{|\mathbf{X_c'}|} \end{bmatrix} \quad \tilde{\mathbf{L}} = \begin{bmatrix} \lambda_{sc} \mathbf{I}_{|\mathbf{X_s}|} & -\lambda_{sc} \tilde{\mathbf{Q}} \\ -\lambda_{sc} \tilde{\mathbf{Q}}^{\mathbf{T}} & \lambda_{sc} \mathbf{I}_{|\mathbf{X_c}|} \end{bmatrix}$$

we can show a **closed form solution** to Equation 7.4 as follows:

$$\hat{\mathbf{w}} = (\tilde{\mathbf{X}}^{\mathbf{T}} \tilde{\mathbf{L}} \tilde{\mathbf{X}} + \tilde{\mathbf{X}}^{\mathbf{T}} \tilde{\mathbf{X}} + \tilde{\mathbf{X}}'^{\mathbf{T}} \tilde{\boldsymbol{\lambda}} \tilde{\mathbf{X}}' + \tilde{\boldsymbol{\beta}})^{-1} (\tilde{\mathbf{X}}^{\mathbf{T}} \tilde{\mathbf{Y}} + \tilde{\mathbf{X}}'^{\mathbf{T}} \tilde{\boldsymbol{\lambda}} \tilde{\mathbf{Y}}') \tag{7.5}$$

7.4 Timeline Generation

Now we present an optimization framework for timeline generation. Formally, for each day, our system takes as input a set of sentences V_s and a set of comments V_c to be summarized, and the (automatically generated) timeline $\mathcal{T}$ (represented as threads) for days prior to the current day. It then identifies a subset $S \subseteq V_s$ as the article summary and a subset $C \subseteq V_c$ as the comment summary by maximizing the following function:

$$\mathcal{Z}(S, C; \mathcal{T}) = \mathcal{S}_{qual}(S; \mathcal{T}) + \mathcal{C}_{qual}(C) + \delta \mathcal{X}(S, C) \tag{7.6}$$

where $\mathcal{S}_{qual}(S; \mathcal{T})$ measures the quality of the article summary S in the context of the historical timeline represented as event threads $\mathcal{T}$; $\mathcal{C}_{qual}(C)$ computes the quality of the comment summary C; and $\mathcal{X}(S, C)$ estimates the connectivity between S and C.

We solve this maximization problem using an alternating optimization algorithm which is outlined in Section 7.4.4. In general, we alternately search for a better article summary S with hill climbing search and a better comment summary C with Ford-Fulkerson algorithm until convergence.

In the rest of this section, we first describe an *entity-centered event threading* algorithm to construct event threads $\mathcal{T}$ which are used to boost article timeline continuity.

Then we explain how to compute $\mathcal{S}_{qual}(S;\mathcal{T})$ and $C_{qual}(C)$ in Section 7.4.2, followed by $\mathcal{X}(S,C)$ in Section 7.4.3.

7.4.1 Entity-Centered Event Threading

We present an event threading process where each thread connects sequential events centered on a set of relevant *entities*. For instance, the following thread connects events about *Obama*'s action towards the annexation of Crimea by *Russia*:

Day 1: *Obama* declared sanctions on *Russian officials*.
Day 2: *President Obama* warned *Russian*.
Day 3: *Obama* urges *Russian* to move back its troops.
Day 4: *Obama* condemns *Russian* aggression in Ukraine.

We first collect relation extractions as *(entity, relation, entity)* triples from OLLIE [Mausam et al., 2012], a dependency relation based open information extraction system. We retain extractions with confidence scores higher than 0.5. We further design syntactic patterns based on Fader et al. [2011] to identify relations expressed as a combination of a verb and nouns. Each relation contains at least one event-related word [Ritter et al., 2012].

The *entity-centered event threading* algorithm works as follows: on the first day, each sentence in the summary becomes an individual cluster; thereafter, each sentence in the current day's article summary either gets attached to an existing thread or starts a new thread. The updated threads then become the input to next day's summary generation process. On day n, we have a set of threads $\mathcal{T} = \{\tau : \mathbf{s_1}, \mathbf{s_2}, \cdots, \mathbf{s_{n-1}}\}$ constructed

from previous $n-1$ days, where $\mathbf{s_i}$ represents the set of sentences attached to thread τ from day i. The *cohesion* between a new sentence $s \in S$ and a thread τ is denoted as $cohn(s, \tau)$. s is attached to $\hat{\tau}$ if there exists $\hat{\tau} = \max_{\tau \in T} cohn(s, \tau)$ and $cohn(s, \hat{\tau}) > 0.0$. Otherwise, s becomes a new thread. We define $cohn(s, \tau) = \min_{\mathbf{s_i} \in \tau, \mathbf{s_i} \neq \emptyset} tfsimi(\mathbf{s_i}, s)$, where $tfsimi(\mathbf{s_i}, s)$ measures the TF similarity between $\mathbf{s_i}$ and s. We consider unigrams/bigrams/trigrams generated from the entities of our event extractions.

7.4.2 Summary Quality Measurement

Recall that we learned two separate importance scoring functions for sentences and comments, which will be denoted here as $imp_s(s)$ and $imp_c(c)$. With an article summary S and threads $\mathcal{T} = \{\tau_i\}$, the **article summary quality** function $\mathcal{S}_{qual}(S; \mathcal{T})$ has the following form:

$$\begin{aligned} \mathcal{S}_{qual}(S; \mathcal{T}) = & \sum_{s \in S} imp(s) \\ & + \theta_{cov} \sum_{s' \in V_s} \min(\sum_{s \in S} tfidf(s, s'), \alpha \sum_{\hat{s} \in V_s} tfidf(\hat{s}, s')) \\ & + \theta_{cont} \sum_{\tau \in \mathcal{T}} \max_{s_k \in S} cohn(s_k, \tau) \end{aligned} \tag{7.7}$$

where $tfidf(\cdot, \cdot)$ is the TF-IDF similarity function. $\mathcal{S}_{qual}(S; \mathcal{T})$ captures three desired qualities of an article summary: *importance* (first item), *coverage* (second item), and the *continuity* of the current summary to previously generated summaries. The coverage function has been used to encourage summary diversity and reduce redundancy [Lin and Bilmes, 2011, Wang et al., 2014]. The continuity function considers how well article summary S can be attached to each event thread, thus favors summaries that can be connected to multiple threads.

Parameters θ_{cov} and α are tuned on multi-document summarization dataset DUC

2003 [Over and Yen, 2003]. Experiments show that system performance peaks and is stable for $\theta_{cont} \in [1.0, 5.0]$. We thus fix θ_{cont} to 1.0. We discard sentences with more than 80% of content words covered by historical summaries. We use BASIC to denote a system that only optimizes on importance and coverage (i.e. first two items in $\mathcal{S}_{qual}(S;\mathcal{T})$). The system optimizing $\mathcal{S}_{qual}(S;\mathcal{T})$ is henceforth called THREAD.

The **comment summary quality** function simply takes the form $C_{qual}(C) = \sum_{c \in C} imp_c(c)$.

7.4.3 Connectivity Measurement

We encode two objectives in the connectivity function $\mathcal{X}(S, C)$: (1) encouraging topical cohesion (i.e. connectivity) between article summary and comment summary; and (2) favoring comments that cover diversified events.

Let $conn(s, c)$ measure content similarity between a sentence $s \in S$ and a comment $c \in C$. Connectivity between article summary S and comment summary C is computed as follows. We build a bipartite graph $\mathcal{G}$ between S and C with edge weight as $conn(s, c)$. We then find an edge set $\mathcal{M}$, the best matching of $\mathcal{G}$. $\mathcal{X}(S, C)$ is defined as the sum over edge weights in $\mathcal{M}$, i.e. $\mathcal{X}(S, C) = \sum_{e \in \mathcal{M}} weight(e)$. An example is illustrated in Figure 7.2.

We consider two options for $conn(s, c)$. One is *lexical similarity* which is based on TF-IDF vectors. Another is *semantic similarity*. Let $R_s = \{(a_s, r_s, b_s)\}$ and $R_c = \{(a_c, r_c, b_c)\}$ be the sets of dependency relations in s and c. $conn(s, c)$ is calculated as:

$$\sum_{(a_s,r_s,b_s)\in R_s} \max_{\substack{(a_c,r_c,b_c)\in R_c \\ r_s=r_c}} simi(a_s, a_c) \times simi(b_s, b_c)$$

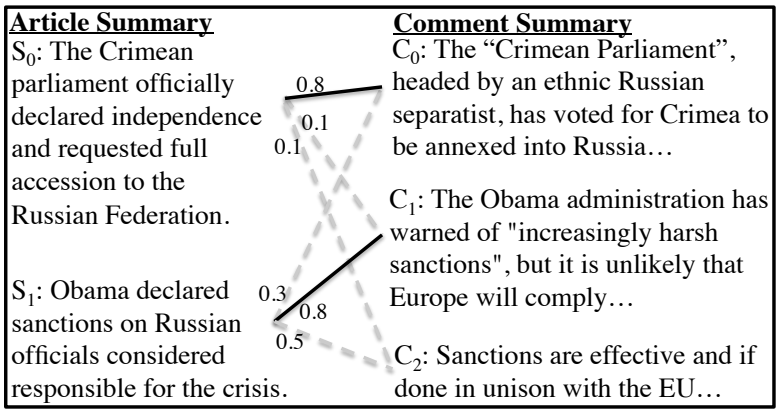


Figure 7.2: An example on computing the connectivity between an article summary (left) and a comment summary (right) via best matching in bipartite graph. Number on each edge indicates the content similarity between a sentence and a comment. Solid lines are edges in the best matching graph. For this example, the connectivity $\mathcal{X}(S, C)$ is $0.8 + 0.8 = 1.6$.

where $simi(\cdot, \cdot)$ is a word similarity function. We experiment with shortest path based similarity defined on WordNet [Miller, 1995] and Cosine similarity with word vectors trained on Google news [Mikolov et al., 2013]. Systems using the three metrics that optimize $\mathcal{Z}(S, C; \mathcal{T})$ are henceforth called THREAD+OPT$_{\texttt{TFIDF}}$, THREAD+OPT$_{\texttt{WordNet}}$ and THREAD+OPT$_{\texttt{WordVec}}$.

7.4.4 An Alternating Optimization Algorithm

To maximize the full objective function $\mathcal{Z}(S, C; \mathcal{T})$, we design a novel alternating optimization algorithm (Alg. 4) where we alternately find better S and C.

We initialize S_0 by a greedy algorithm [Lin and Bilmes, 2011] with respect to $\mathcal{S}_{qual}(S; \mathcal{T})$. Notice that $\mathcal{S}_{qual}(S; \mathcal{T})$ is a submodular function, so that the greedy solution is a $1 - 1/e$ approximation to the optimal solution of $\mathcal{S}_{qual}(S; \mathcal{T})$. Fixing S_0, we model the problem of finding C_0 that maximizes $\mathcal{C}_{qual}(C) + \delta\mathcal{X}(S_0, C)$ as a maximum-weight bipartite graph matching problem. This problem can be reduced to a maximum

network flow problem, and then be solved by Ford-Fulkerson algorithm (details are discussed in Kleinberg and Tardos [2005]). Thereafter, for each iteration, we alternately find a better S_t with regard to $\mathcal{S}_{qual}(S;\mathcal{T}) + \delta\mathcal{X}(S, C_{t-1})$ using hill climbing, and an exact solution C_t to $C_{qual}(C) + \delta\mathcal{X}(S_t, C)$ with Ford-Fulkerson algorithm. Iteration stops when the increase of $\mathcal{Z}(S, C)$ is below threshold ϵ (set to 0.01). System performance is stable when we vary $\delta \in [1.0, 10.0]$, so we set $\delta = 1.0$.

Input : sentences V_s, comments V_c, threads $\mathcal{T}$, δ, threshold ϵ, functions $\mathcal{Z}(S, C; \mathcal{T})$, $\mathcal{S}_{qual}(S;\mathcal{T})$, $C_{qual}(C)$, $\mathcal{X}(S, C)$
Output: article summary S, comment summary C

```
/* Initialize S and C by greedy algorithm and
   Ford-Fulkerson algorithm                          */
```
$S_0 \leftarrow \max_S \mathcal{S}_{qual}(S;\mathcal{T})$;
$C_0 \leftarrow \max_C C_{qual}(C) + \delta\mathcal{X}(S_0, C)$;
$t \leftarrow 1$;
$\Delta\mathcal{Z} \leftarrow \infty$;
while $\Delta\mathcal{Z} > \epsilon$ **do**
```
    /* Step 1:  Hill climbing algorithm              */
```
    $S_t \leftarrow \max_S \mathcal{S}_{qual}(S;\mathcal{T}) + \delta\mathcal{X}(S, C_{t-1})$;
```
    /* Step 2:  Ford-Fulkerson algorithm             */
```
    $C_t \leftarrow \max_C C_{qual}(C) + \delta\mathcal{X}(S_t, C)$;
    $\Delta\mathcal{Z} = \mathcal{Z}(S_t, C_t; \mathcal{T}) - \mathcal{Z}(S_{t-1}, C_{t-1}; \mathcal{T})$;
    $t \leftarrow t + 1$;
end

Algorithm 4: Generate article summary and comment summary for a given day via alternating optimization.

Algorithm 4 is guaranteed to find a solution at least as good as S_0 and C_0. It progresses only if Step 1 finds S_t that improves upon $\mathcal{Z}(S_{t-1}, C_{t-1}; \mathcal{T})$, and Step 2 finds C_t where $\mathcal{Z}(S_t, C_t; \mathcal{T}) \geq \mathcal{Z}(S_t, C_{t-1}; \mathcal{T})$.

7.5 Experimental Results

7.5.1 Evaluation of SENTENCE and COMMENT Importance Scorers

We test importance scorers (Section 7.3) on single document *sentence ranking* and *comment ranking*.

For both tasks, we compare with two previous systems on joint ranking and summarization of news articles and tweets. *Yang et al. [2011]* employ supervised learning based on factor graphs to model content similarity between the two types of data. We use the same features for this model. *Gao et al. [2012]* summarize by including the complementary information between articles and tweets, which is estimated by an unsupervised topic model.[3] We also consider two state-of-the-art rankers: *RankBoost* [Freund et al., 2003] and *LambdaMART* [Burges, 2010]. Finally, we use a *position baseline* that ranks sentences based on their position in the article, and a *rating baseline* that ranks comments based on positive user ratings.

We evaluate using normalized discounted cumulative gain at top 3 returned results (NDCG@3). Sentences are considered relevant if they have ROUGE-2 scores larger than 0.0 (computed against human abstracts), and comments are considered relevant if they are editor's picks.[4] Figure 7.3 demonstrates that our joint learning model uniformly outperforms all the other comparisons for both ranking tasks. In general, supervised learning based approaches (e.g. our method, Yang et al. [2011], RankBoost, and LambdaMART) produce better results than unsupervised method (e.g. Gao et al. [2012]).

[3] We thank Zi Yang and Peng Li for providing the code.

[4] We experiment with all articles for sentence ranking, and NYT comments (with editor's picks) for comment ranking.

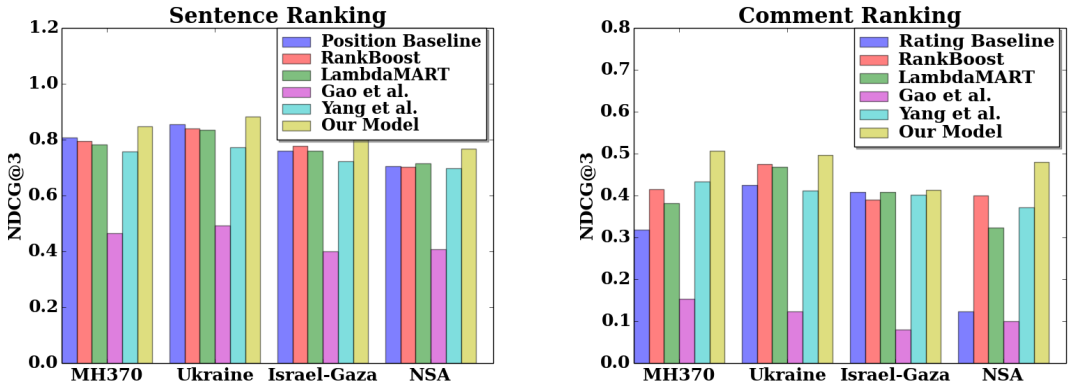


Figure 7.3: Evaluation of sentence and comment ranking on the four datasets by using normalized discounted cumulative gain at top 3 returned results (NDCG@3). Our joint learning based approach uniformly outperforms all the other comparisons.

7.5.2 Leveraging User Comments

In this section, we test if our system can leverage comments to produce better article-based summaries for event timelines. We collect **gold-standard timelines** for each of the four events from the corresponding Wikipedia page(s), NYT topic page, or BBC news page.

We consider two existing timeline creation systems that only utilize news articles, and a timeline generated from single-article human abstracts: (1) CHIEU AND LEE [2004] select sentences with high "interestingness" and "burstiness" using a likelihood ratio test to compare word distributions of sentences with articles in neighboring days. (2) YAN ET AL. [2011] design an evolutionary summarization system that selects sentences based on on coverage, coherence, and diversity. (3) We construct a timeline from the human ABSTRACTs provided with each article: we sort them chronologically according to article timestamps and add abstract sentences into each daily summary until reaching the word limit.

We test on five variations of our system. The first two systems generate ar-

ticle summaries with no comment information by optimizing $S_{qual}(S;\mathcal{T})$ using a greedy algorithm: BASIC ignores event threading; THREAD considers the threads. THREAD+OPT$_{\texttt{TFIDF}}$, THREAD+OPT$_{\texttt{WordNet}}$ and THREAD+OPT$_{\texttt{WordVec}}$ (see Section 7.4.3) leverage user comments to generate article summaries as well as comment summaries based on alternating optimization of Equation 3. Although comment summaries are generated, they are not used in the evaluation.

For all systems, we generate daily article summaries of at most 100 words, and select 5 comments for the corresponding comment summary. We employ ROUGE [Lin and Hovy, 2003] to automatically evaluate the content coverage (in terms of ngrams) of the article-based timelines vs. gold-standard timelines. ROUGE-2 (measures bigram overlap) and ROUGE-SU4 (measures unigram and skip-bigrams separated by up to four words) scores are reported in Table 7.4. As can be seen, under the alternating optimization framework, our systems, employing both articles and comments, consistently yield better ROUGE scores than the three baseline systems and our systems that do not leverage comments. Though constructed from single-article abstracts, baseline ABSTRACT is found to contain redundant information and thus limited in content coverage. This is due to the fact that different media tend to report on the same important events.

7.5.3 Evaluating Socially-Informed Timelines

We evaluate the full article+comment-based timelines on Amazon Mechanical Turk. Turkers are presented with a timeline consisting of five consecutive days' article summaries and four variations of the accompanying comment summary: RANDOMly selected comments, USER'S-PICKS (ranked by positive user ratings), *randomly* selected EDITOR'S-PICKS and timelines produced by the THREAD+OPT$_{\texttt{WordVec}}$ version of OUR

	MH370		Ukraine		Israel-Gaza		NSA	
	R-2	*R-SU4*	*R-2*	*R-SU4*	*R-2*	*R-SU4*	*R-2*	*R-SU4*
CHIEU AND LEE	6.43	10.89	4.64	8.87	3.38	**7.32**	6.14	9.73
YAN ET AL.	6.37	10.35	4.57	8.67	2.39	5.78	3.99	7.73
ABSTRACT	6.16	10.62	3.85	8.40	2.21	5.42	7.03	8.65
- Greedy Algorithm								
BASIC	6.59	9.80	5.31	9.23	3.15	6.20	3.81	7.58
THREAD	6.55	10.86	5.73	9.75	3.16	6.16	6.29	10.09
- Alternating Optimization (leveraging comments)								
THREAD+OPT$_{\texttt{TFIDF}}$	*8.74*	11.63	*9.10*	*12.59*	3.78	6.45	*8.07*	*10.31*
THREAD+OPT$_{\texttt{WordNet}}$	*8.73*	***11.87***	*8.67*	*12.10*	**4.11**	6.64	***8.63***	***11.12***
THREAD+OPT$_{\texttt{WordVec}}$	***9.29***	11.63	***9.16***	***12.72***	3.75	6.38	*8.29*	*10.36*

Table 7.4: ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) scores (multiplied by 100) for different timeline generation approaches on four event datasets. Systems that statistically significantly outperform the three baselines ($p < 0.05$, paired $t-$test) are in *italics*. Numbers in **bold** are the highest score for each column.

SYSTEM. We also include one noisy comment summary (i.e. irrelevant to the question) to avoid spam. We display two comments per day for each system.[5]

Turkers are asked to rank the comment summary variations according to *informativeness* and *insightfulness*. For informativeness, we ask the Turkers to judge based only on knowledge displayed in the timeline, and to rate each comment summary based on how much relevant information they learn from it. For insightfulness, Turkers are required to focus on insights and valuable opinions. They are requested to leave a short explanation of their ranking.

15 five-day periods are randomly selected. We solicit four distinct Turkers located in the U.S. to evaluate each set of timelines. An inter-rater agreement of Krippendorff's α of 0.63 is achieved for informativeness ranking and α is 0.50 for insightfulness ranking.

Table 7.5 shows the percentage of times a particular method is selected as producing the best comment portion of the timeline, as well as the micro-average rank of each

[5]For our system, we select the two comments with highest importance scores from the comment summary.

	Informativeness		Insightfulness	
	% Best	Avg Rank	% Best	Avg Rank
Random	1.7%	3.67	3.3%	3.58
User's-picks	5.0%	2.83	15.0%	2.55
Editor's-picks	26.7%	2.05	30.0%	2.22
Our system	**66.7%**	**1.45**	**51.7%**	**1.65**

Table 7.5: Human evaluation results on the comment portion of socially-informed timelines. **Boldface** indicates statistical significance vs. other results in the same column using a Wilcoxon signed-rank test ($p < 0.05$). On average, the output from our system is ranked higher than all other alternatives.

method, for both informativeness and insightfulness. Our system is selected as the best in 66.7% of the evaluations for informativeness and 51.7% for insightfulness. In both cases, we statistically significantly outperform ($p < 0.05$ using a Wilcoxon signed-rank test) the editor's-picks and user's-picks. Turkers' explanations indicate that they prefer our comment summaries mainly because they are "very informative and insightful to what was happening", and "show the sharpness of the commenter". Turkers sometimes think the summaries randomly selected from editor's-picks "lack connection", and characterize user's-picks as "the information was somewhat limited".

Figure 7.4 shows part of the timeline generated by our system for the Ukraine crisis.

7.5.4 Human Evaluation of Event Threading

Here we evaluate on the utility of event threads for high-level information access guidance: *can event threads allow users to easily locate and absorb information with a specific interest in mind?*

We first sample a 10-day timeline for each dataset from those produced by the THREAD+OPT$_{\texttt{WordVec}}$ variation of our system. We designed one question for each timeline. Sample questions are: "describe the activities for searching for the missing

Article Summary	Comment Summary
2014-03-17 Obama administration froze the U.S. assets of seven Russian officials, while similar sanctions were imposed on four Ukrainian officials. . . .	Theodore Roosevelt said that the worst possible thing you can do in diplomacy is "soft hitting". That is what the US and the EU are doing in these timid "sanctions" against people without any overseas accounts. . .
2014-03-18 Ukraine does not recognize a treaty signed in Moscow on Tuesday making its Crimean peninsula a part of Russia. . .	Though there were many in Crimea who supported annexation, there were certainly some who did not. what about those people?. . .
2014-03-19 The head of NATO warned on Wednesday that Russian President Vladimir Putin may not stop with the annexation of Crimea . . .	If you look at a real map , Crimea is an island and has always been more connected to Russia than to Ukraine. . .
2014-03-20 The United States on Thursday expanded its sanctions on Russians. . . in response to the annexation of Crimea . . .	The US and EU should follow up economic sanctions with concrete steps to strengthen NATO. . .

Figure 7.4: A snippet of timeline generated by our system THREAD+OPT$_{\texttt{WordVec}}$ for the Ukraine crisis.

flight MH370", and "describe the attitude and action of Russian Government on Eastern Ukraine". We recruited 10 undergraduate and graduate students who are native speakers of English. Each student first read one question and its corresponding timeline for 5 minutes. The timeline was then removed, and the student wrote down an answer for the question. We asked each student to answer the question for each of four timelines (one for each event dataset). Two timelines are displayed with threads, and two without threads. We presented threads by adding a thread number in front of each sentence.

We then used Amazon Mechanical Turk to evaluate the informativeness of students' answers. Turkers were asked to read all 10 answers for the same question, with five answers based on timelines with threads and five others based on timelines without threads. After that, they rated each answer with an informativeness score on a 1-to-5 rating scale (1 as "not relevant to the query", and 5 as "very informative"). We also

added two quality control questions. Table 7.6 shows that the average rating for answers written after reading timelines *with threads* is 3.29 (43% are rated ≥ 4), higher than the 2.58 for the timelines with *no thread* exhibited (30% are rated ≥ 4).

Answer Type	**Avg ± STD**	**Rated 5 (%)**	**Rated 4 (%)**
No Thread	2.58 ± 1.20	7%	23%
With Threads	3.29 ± 1.28	17%	26%

Table 7.6: Human evaluation on the informativeness of answers written after reading timelines *with threads* vs. with *no thread*. Answers written with access to threads are rated higher (3.29) than the ones with no thread (2.58).

7.6 Conclusion

We presented a socially-informed timeline generation system that constructs timelines consisting of article summaries and comment summaries. An alternating optimization algorithm is designed to maximize the connectivity between the two sets of summaries as well as their importance and information coverage. Automatic and human evaluations showed that our system produced more informative timelines than state-of-the-art systems. Our comment summaries were also rated as very insightful.

CHAPTER 8

SENTIMENT ANALYSIS FOR ONLINE SOCIAL INTERACTION

In this chapter, we present our work that contributes to utilizing sentiment analysis techniques to study online social interactions.

8.1 Introduction

We are in an era where people can easily voice and exchange their opinions on the internet through forums or social media. Mining public opinion and the social interactions from online discussions is an important task, which has a wide range of applications. For example, by analyzing the users' attitude in forum posts on social and political problems, it is able to identify ideological stance [Somasundaran and Wiebe, 2009] and user relations [Qiu et al., 2013], and thus further discover subgroups [Hassan et al., 2012, Abu-Jbara et al., 2012] with similar ideological viewpoint. Meanwhile, catching the sentiment in the conversation can help detect online disputes, reveal popular or controversial topics, and potentially disclose the public opinion formation process.

In this chapter, we first study the problem of agreement and disagreement identification in online discussions (see Section 8.2). Sentence-level agreement and disagreement detection for this domain is challenging in its own right due to the dynamic nature of online conversations, and the less formal, and usually very emotional language used. As an example, consider a snippet of discussion from Wikipedia Talk page for article "Iraq War" where editors argue on the correctness of the information in the opening paragraph (Figure 8.1). *"So what?"* should presumably be tagged as a negative sentence as should the sentence *"If you're going to troll, do us all a favor and stick to the guidelines."*. We hypothesize that these, and other, examples will be difficult for the tagger unless

Zer0faults: So questions comments feedback welcome. Other views etc. I just hope we can remove the assertations that WMD's were in fact the sole reason for the US invasion, considering that HJ Res 114 covers many many reasons.
>**Mr. Tibbs**: So basically what you want to do is remove all mention of the cassus belli of the Iraq War and try to create the false impression that this military action was as inevitable as the sunrise.$_{[NN]}$ No. Just because things didn't turn out the way the Bush administration wanted doesn't give you license to rewrite history.$_{[NN]}$ ...
>>**MONGO**: Regardless, the article is an antiwar propaganda tool.$_{[NN]}$ ...
>>>**Mr. Tibbs**: So what?$_{[NN]}$ That wasn't the cassus belli and trying to give that impression After the Fact is Untrue.$_{[NN]}$ Hell, the reason it wasn't the cassus belli is because there are dictators in Africa that make Saddam look like a pussycat...
>>**Haizum**: Start using the proper format or it's over for your comments.$_{[N]}$ If you're going to troll, do us all a favor and stick to the guidelines.$_{[N]}$ ...
Tmorton166: Hi, I wonder if, as an outsider to this debate I can put my word in here. I considered mediating this discussion however I'd prefer just to comment and leave it at that :). I agree mostly with what Zer0faults is saying$_{[PP]}$. ...
>**Mr. Tibbs**: Here's the problem with that.$_{[NN]}$ It's not about publicity or press coverage. It's about the fact that the Iraq disarmament crisis set off the 2003 Invasion of Iraq. ... And theres a huge problem with rewriting the intro as if the Iraq disarmament crisis never happened.$_{[NN]}$
>>**Tmorton166**: ... To suggest in the opening paragraph that the ONLY reason for the war was WMD's is wrong - because it simply isn't.$_{[NN]}$ However I agree that the emphasis needs to be on the armaments crisis because it was the reason sold to the public and the major one used to justify the invasion but it needs to acknowledge that there was at least 12 reasons for the war as well.$_{[PP]}$ ...

Figure 8.1: Example discussion from wikipedia talk page for article "Iraq War", where editors discuss about the correctness of the information in the opening paragraph. We only show some sentences that are relevant for demonstration. Other sentences are omitted by ellipsis. Names of editors are in **bold**. ">" is an indicator for the reply structure, where turns starting with > are response for most previous turn that with one less >. We use "*NN*", "*N*", and "*PP*" to indicate "strongly disagree", "disagree", and "strongly agree". Sentences in blue are examples whose sentiment is hard to detect by an existing lexicon.

the context surrounding each sentence is considered and in the absence of a sentiment lexicon tuned for conversational text [Ding et al., 2008, Choi and Cardie, 2009].

As a result, we investigate isotonic Conditional Random Fields (isotonic CRF) [Mao and Lebanon, 2007] for the sentiment tagging task since they preserve the advantages of the popular CRF sequential tagging models [Lafferty et al., 2001a] while providing an

efficient mechanism to encode domain knowledge — in our case, a sentiment lexicon — through isotonic constraints on the model parameters. We employ two existing online discussion data sets: the *Authority and Alignment in Wikipedia Discussions (AAWD)* corpus of Bender et al. [2011] (Wikipedia talk pages) and the *Internet Argument Corpus (IAC)* of Walker et al. [2012a]. Experimental results show that our model significantly outperforms state-of-the-art methods on the AAWD data (our F1 scores are 0.74 and 0.67 for agreement and disagreement, vs. 0.58 and 0.56 for the linear chain CRF approach) and IAC data (our F1 scores are 0.61 and 0.78 for agreement and disagreement, vs. 0.28 and 0.73 for SVM). In particular, we bootstrap the construction of a sentiment lexicon from Wikipedia talk pages using the lexical items in existing general-purpose sentiment lexicons as seeds and in conjunction with an existing label propagation algorithm [Zhu and Ghahramani, 2002].

In Section 8.3, we will show how to use the agreement and disagreement classifier to detect the sentiment flows in online discussions and study the task of *dispute detection*. As the web has grown in popularity and scope, so has the promise of collaborative information environments for the joint creation and exchange of knowledge [Jones and Rafaeli, 2000, Sack, 2005]. Wikipedia, a wiki-based online encyclopedia, is arguably the best example: its distributed editing environment allows readers to collaborate as content editors and has facilitated the production of over four million articles[1] of surprisingly high quality [Giles, 2005] in English alone since its debut in 2001.

Existing studies of collaborative knowledge systems have shown, however, that the quality of the generated content (e.g. an encyclopedia article) is highly correlated with the effectiveness of the online collaboration [Kittur and Kraut, 2008, Kraut and Resnick, 2012]; fruitful collaboration, in turn, inevitably requires dealing with the disputes and conflicts that arise [Kittur et al., 2007]. Unfortunately, human monitoring of the often

[1] `http://en.wikipedia.org`

massive social media and collaboration sites to detect, much less mediate, disputes is not feasible.

Previous work has analyzed dispute-laden content to discover features correlated with conflicts and disputes [Kittur et al., 2007]. Research focused primarily on cues derived from the edit history of the jointly created content (e.g. the number of revisions, their temporal density [Kittur et al., 2007, Yasseri et al., 2012]) and relied on normalsize numbers of manually selected discussions known to involve disputes. In contrast, we investigate methods for the automatic detection, i.e. prediction, of discussions involving disputes. Though Mishne and Glance [2006] studied automatic detection of disputed comment threads in weblogs, they experimented with a dataset of small scale. We are also interested in understanding whether, and which, linguistic features of the discussion are important for dispute detection.

Drawing inspiration from studies of human mediation of online conflicts (e.g. Billings and Watts [2010], Kittur et al. [2007], Kraut and Resnick [2012]), we hypothesize that effective methods for dispute detection should take into account the sentiment and opinions expressed by participants in the collaborative endeavor. As a result, we propose a sentiment analysis approach for online dispute detection that identifies the sequence of sentence-level sentiments (i.e. very negative, negative, neutral, positive, very positive) expressed during the discussion and uses them as features in a classifier that predicts the DISPUTE/NON-DISPUTE label for the discussion as a whole. Consider, for example, the snippet in Figure 8.2 from the Wikipedia Talk page for the article on Philadelphia; it discusses the choice of a picture for the article's "infobox". The sequence of almost exclusively negative statements provides evidence of a dispute in this portion of the discussion.

Unfortunately, sentence-level sentiment tagging for this domain is challenging in its

1-**Emy111**: I think everyone is forgetting that my previous image was the lead image for well over a year! ...
> **Massimo**: I'm sorry to say so, but it is grossly over processed...
2-**Emy111**: i'm glad you paid more money for a camera than I did. congrats... i appreciate your constructive criticism. thank you.
> **Massimo**: I just want to have the best picture as a lead for the article ...
3-**Emy111**: Wow, I am really enjoying this photography debate... [so don't make assumptions you know nothing about.]$_{\mathbf{NN}}$ [Really, grow up.]$_{\mathbf{N}}$ [If you all want to complain about Photoshop editing, lets all go buy medium format film cameras, shoot film, and scan it, so no manipulation is possible.]$_{\mathbf{O}}$ [Sound good?]$_{\mathbf{NN}}$
> **Massimo**: ... I do feel it is a pity, that you turned out to be a sore loser...

Figure 8.2: From the Wikipedia Talk page for the article "Philadelphia". Omitted sentences are indicated by ellipsis. Names of editors are in **bold**. The start of each set of related turns is numbered; ">" is an indicator for the reply structure.

own right due to the less formal, often ungrammatical, language and the dynamic nature of online conversations. "*Really, grow up*" (segment 3) should presumably be tagged as a negative sentence as should the sarcastic sentences "*Sounds good?*" (in the same turn) and "*congrats*" and "*thank you*" (in segment 2). We expect that these, and other, examples will be difficult for the sentence-level classifier unless the discourse context of each sentence is considered. Previous research on sentiment prediction for online discussions, however, focuses on turn-level predictions [Hahn et al., 2006, Yin et al., 2012].[2] As the first work that predicts sentence-level sentiment for online discussions, we investigate isotonic Conditional Random Fields (CRFs) [Mao and Lebanon, 2007] for the sentiment-tagging task as they preserve the advantages of the popular CRF-based sequential tagging models [Lafferty et al., 2001a] while providing an efficient mechanism for encoding domain knowledge — in our case, a sentiment lexicon — through isotonic constraints on model parameters.

We evaluate our dispute detection approach using a newly created corpus of dis-

[2]A notable exception is Hassan et al. [2010], which identifies sentences containing "attitudes" (e.g. opinions), but does not distinguish them w.r.t. sentiment. Context information is also not considered.

cussions from Wikipedia Talk pages (3609 disputes, 3609 non-disputes).[3] We find that classifiers that employ the learned sentiment features outperform others that do not. The best model achieves a very promising F1 score of 0.78 and an accuracy of 0.80 on the Wikipedia dispute corpus. To the best of our knowledge, this represents the first computational approach to automatically identify online disputes on a dataset of scale.

8.2 Agreement and Disagreement Identification in Online Discussions

8.2.1 The Model

We first give a brief overview on isotonic Conditional Random Fields (isotonic CRF) [Mao and Lebanon, 2007], which is used as the backbone approach for our sentence- or segment-level agreement and disagreement detection model. We defer the explanation of online discussion lexicon construction in Section 8.2.2.

Problem Description

Consider a discussion comprised of sequential turns uttered by the participants; each turn consists of a sequence of text units, where each unit can be a sentence or a segment of several sentences. Our model takes as input the text units $\mathbf{x} = \{x_1, \cdots, x_n\}$ in the same turn, and outputs a sequence of sentiment labels $\mathbf{y} = \{y_1, \cdots, y_n\}$, where $y_i \in \mathcal{O}, \mathcal{O} = \{\text{NN}, \text{N}, \text{O}, \text{P}, \text{PP}\}$. The labels in $\mathcal{O}$ represent strongly disagree (NN), disagree

[3]The talk page associated with each article records conversations among editors about the article content and allows editors to discuss the writing process, e.g. planning and organizing the content.

(N), neutral (O), agree (P), strongly agree (PP), respectively. In addition, elements in the partially ordered set O possess an ordinal relation $\leq$. Here, we differentiate agreement and disagreement with different intensity, because the output of our classifier can be used for other applications, such as dispute detection, where "strongly disagree" (e.g. NN) plays an important role. Meanwhile, fine-grained sentiment labels potentially provide richer context information for the sequential model employed for this task.

Isotonic Conditional Random Fields

Conditional Random Fields (CRF) have been successfully applied in numerous sequential labeling tasks [Lafferty et al., 2001a]. Given a sequence of utterances or segments $\mathbf{x} = \{x_1, \cdots, x_n\}$, according to linear-chain CRF, the probability of the labels $\mathbf{y}$ for $\mathbf{x}$ is given by:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x}) = & \frac{1}{Z(\mathbf{x})} \exp\Big(\sum_i \sum_{\sigma,\tau} \lambda_{\langle\sigma,\tau\rangle} f_{\langle\sigma,\tau\rangle}(y_{i-1}, y_i) \\ & + \sum_i \sum_{\sigma,w} \mu_{\langle\sigma,w\rangle} g_{\langle\sigma,w\rangle}(y_i, x_i)\Big) \end{aligned} \tag{8.1}$$

$f_{\langle\sigma,\tau\rangle}(y_{i-1}, y_i)$ and $g_{\langle\sigma,w\rangle}(y_i, x_i)$ are feature functions. Given that y_{i-1}, y_i, x_i take values of σ, τ, w, the functions are indexed by pairs $\langle\sigma,\tau\rangle$ and $\langle\sigma, w\rangle$. $\lambda_{\langle\sigma,\tau\rangle}, \mu_{\langle\sigma,w\rangle}$ are the parameters.

CRF, as defined above, is not appropriate for ordinal data like sentiment, because it ignores the ordinal relation among sentiment labels. Isotonic Conditional Random Fields (isotonic CRF) are proposed by Mao and Lebanon [2007] to enforce a set of monotonicity constraints on the parameters that are consistent with the ordinal structure and domain knowledge (in our case, a sentiment lexicon automatically constructed from online discussions).

Given a lexicon $\mathcal{M} = \mathcal{M}_p \cup \mathcal{M}_n$, where $\mathcal{M}_p$ and $\mathcal{M}_n$ are two sets of features (usually words) identified as strongly associated with positive sentiment and negative sentiment. The constraints are encoded as below. For each feature $w \in \mathcal{M}_p$, isotonic CRF enforces $\sigma \leq \sigma' \Rightarrow \mu_{\langle\sigma,w\rangle} \leq \mu_{\langle\sigma',w\rangle}$. Intuitively, the parameters $\mu_{\langle\sigma,w\rangle}$ are intimately tied to the model probabilities. When a feature such as "totally agree" is observed in the training data, the feature parameter for $\mu_{\langle\text{PP,totally agree}\rangle}$ is likely to increase. Similar constraints are also defined on $\mathcal{M}_n$. In this work, we boostrap the construction of an online discussion sentiment lexicon used as $\mathcal{M}$ in the isotonic CRF (see Section 8.2.2).

The parameters can be found by maximizing the likelihood subject to the monotonicity constraints. We adopt the re-parameterization from Mao and Lebanon [2007] for a simpler optimization problem. [4]

Features

The features used in sentiment prediction are listed in Table 8.1. Features with numerical values are first normalized by standardization, then binned into 5 categories.

Syntactic/Semantic Features. Dependency relations have been shown to be effective for various sentiment prediction tasks [Joshi and Penstein-Rosé, 2009, Somasundaran and Wiebe, 2009, Hassan et al., 2010, Abu-Jbara et al., 2012]. We have two versions of dependency relation as features, one being the original form, another generalizing a word to its POS tag in turn. For instance, "nsubj(wrong, you)" is generlized as the "nsubj(`ADJ`, you)" and "nsubj(wrong, `PRP`)". We use Stanford parser [de Marneffe et al., 2006] to obtain parse trees and dependency relations.

[4] The full implementation is based on MALLET **?**. We thank Yi Mao for sharing the implementation of the core learning algorithm.

Lexical Features
- unigram/bigram
- num of words all uppercased
- num of words
Discourse Features
- initial uni-/bi-/trigram
- repeated punctuations
- hedging [Farkas et al., 2010]
- number of negators
Syntactic/Semantic Features
- unigram with POS tag
- dependency relation
Conversation Features
- quote overlap with target
- TFIDF similarity with target (remove quote first)
Sentiment Features
- connective + sentiment words
- sentiment dependency relation
- sentiment words

Table 8.1: Features used in sentiment prediction for online discussions.

Discourse Features. Previous work [Hirschberg and Litman, 1993, Abbott et al., 2011] suggests that discourse markers, such as *what?*, *actually*, may have their use for expressing opinions. We extract the initial unigram, bigram, and trigram of each utterance as discourse features [Hirschberg and Litman, 1993]. Hedge words are collected from the CoNLL-2012 shared task [Farkas et al., 2010].

Conversation Features. Conversation features encode some useful information regarding the similarity between the current utterance(s) and the sentences uttered by the target participant. TFIDF similarity is computed. We also check if the current utterance(s) quotes target sentences and compute its length.

Sentiment Features. We gather connectives from Penn Discourse TreeBank [Prasad et al., 2008] and combine them with any sentiment word that precedes or follows it as

new features. Sentiment dependency relations are the subset of dependency relations with sentiment words. We replace those words with their polarity equivalents. For example, relation "nsubj(wrong, you)" becomes "nsubj($\texttt{SentiWord}_{neg}$, you)".

8.2.2 Online Discussion Sentiment Lexicon Construction

POSITIVE
please elaborate, nod, await response, from experiences, anti-war, profits, promises of, is undisputed, royalty, sunlight, conclusively, badges, prophecies, in vivo, tesla, pioneer, published material, from god, plea for, lend itself, geek, intuition, morning, anti $\texttt{SentiWord}_{neg}$, connected closely, Rel(undertake, to), intelligibility, Rel(articles, detailed), of noting, for brevity, Rel(believer, am), endorsements, testable, source carefully
NEGATIVE
: (, TOT, ?!!, in contrast, ought to, whatever, Rel(nothing, you), anyway, Rel(crap, your), by facts, purporting, disproven, Rel(judgement, our), Rel(demonstrating, you), opt for, subdue to, disinformation, tornado, heroin, Rel(newbies, the), Rel (intentional, is), pretext, watergate, folly, perjury, Rel(lock, article), contrast with, poke to, censoring information, partisanship, insurrection, bigot, Rel(informative, less), clowns, Rel(feeling, mixed), never-ending

Table 8.2: Example terms and relations from our online discussion lexicon. We choose for display terms that do not contain any seed word.

So far as we know, there is no lexicon available for online discussions. Thus, we create from a large-scale corpus via *label propagation*. The label propagation algorithm, proposed by Zhu and Ghahramani [2002], is a semi-supervised learning method. In general, it takes as input a set of seed samples (e.g. sentiment words in our case), and the similarity between pairwise samples, then iteratively assigns values to the unlabeled samples (see Algorithm 5). The construction of graph G is discussed in the next section. Sample sentiment words in the new lexicon are listed in Table 8.2.

Input : $G = (V, E), w_{ij} \in [0, 1]$, positive seed words P, negative seed words N, number of iterations T

Output: $\{y_i\}_{i=0}^{|V|-1}$

$y_i = 1.0, \forall v_i \in P$

$y_i = -1.0, \forall v_i \in N$

$y_i = 0.0, \forall v_i \notin P \cup N$

for $t = 1 \cdots T$ **do**

$y_i = \frac{\sum_{(v_i, v_j) \in E} w_{ij} \times y_j}{\sum_{(v_i, v_j) \in E} w_{ij}}, \forall v_i \in V$

$y_i = 1.0, \forall v_i \in P$

$y_i = -1.0, \forall v_i \in N$

end

Algorithm 5: The label propagation algorithm [Zhu and Ghahramani, 2002] used for constructing online discussion lexicon.

Graph Construction

Node Set V. Traditional lexicons, like General Inquirer [Stone et al., 1966], usually consist of polarized unigrams. As we mentioned in Section 8.1, unigrams lack the capability of capturing the sentiment conveyed in online discussions. Instead, bigrams, dependency relations, and even punctuation can serve as supplement to the unigrams. Therefore, we consider four types of *text units* as nodes in the graph: unigrams, bigrams, dependency relations, sentiment dependency relations. Sentiment dependency relations are described in Section 8.2.1. We replace all relation names with a general label. Text units that appear in at least 10 discussions are retained as nodes to reduce noise.

Edge Set E. As Velikovich et al. [2010] and Feng et al. [2013] notice, a dense graph with a large number of nodes is susceptible to propagating noise, and will not scale well. We thus adopt the algorithm in Feng et al. [2013] to construct a sparsely connected graph. For each text unit t, we first compute its representation vector $\vec{d}$ using Pairwise Mutual Information scores with respect to the top 50 co-occuring text units. We define "co-occur" as text units appearing in the same sentence. An edge is created between

two text units t_0 and t_1 only if they ever co-occur. The similarity between t_0 and t_1 is calculated as the Cosine similarity between $\vec{a}_0$ and $\vec{a}_1$.

Seed Words. The seed sentiment are collected from three existing lexicons: MPQA lexicon, General Inquirer, and SentiWordNet. Each word in SentiWordNet is associated with a positive score and a negative score; words with a polarity score larger than 0.7 are retained. We remove words with conflicting sentiments.

Data

The graph is constructed based on Wikipedia talk pages. We download the 2013-03-04 Wikipedia data dump, which contains 4,412,582 talk pages. Since we are interested in conversational languages, we filter out talk pages with fewer than 5 participants. This results in a dataset of 20,884 talk pages, from which the graph is constructed.

8.2.3 Experimental Setup

Datasets

Wikipedia Talk pages. The first dataset we use is *Authority and Alignment in Wikipedia Discussions (AAWD)* corpus [Bender et al., 2011]. AAWD consists of 221 English Wikipedia discussions with agreement and disagreement annotations.[5]

The annotation of AAWD is made at utterance- or turn-level, where a turn is defined as continuous body of text uttered by the same participant. Annotators either label each

[5]Bender et al. [2011] originally use positive alignment and negative alignment to indicate two types of social moves. They define those alignment moves as "agreeing or disagreeing" with the target. We thus use agreement and disagreement instead of positive and negative alignment in this work.

utterance as agreement, disagreement or neutral, and select the corresponding spans of text, or label the full turn. Each turn is annotated by two or three people. To induce an utterance-level label for instances that have only a turn-level label, we assume they have the same label as the turn.

To train our sentiment model, we further transform agreement and disagreement labels (i.e. 3-way) into the 5-way labels. For utterances that are annotated as agreement and have the text span specified by at least two annotators, they are treated as "strongly agree" (PP). If an utterance is only selected as agreement by one annotator or it gets the label by turn-level annotation, it is "agree" (P). "Strongly disagree" (NN) and "disagree" (N) are collected in the same way from disagreement label. All others are neutral (O). In total, we have 16,501 utterances. 1,930 and 1,102 utterances are labeled as "NN" and "N". 532 and 99 of them are "PP" and "P". All other 12,648 are neutral samples. [6]

Online Debate. The second dataset is the *Internet Argument Corpus (IAC)* [Walker et al., 2012a] collected from an online debate forum. Each discussion in IAC consists of multiple posts, where we treat each post as a turn. Most posts (72.3%) contain quoted content from the posts they target at or other resources. A post can have more than one quote, which naturally break the post into multiple segments. 1,806 discussions are annotated with agreement and disagreement on the segment-level from -5 to 5, with -5 as strongly disagree and 5 as strongly agree. We first compute the average score for each segment among different annotators and transform the score into sentiment label in the following way. We treat $[-5, -3]$ as NN (1595 segments), $(-3, -1]$ as N (4548 segments), $[1, 3)$ as P (911 samples), $[3, 5]$ as PP (199), all others as O (2701 segments).

In the test phase, utterances or segments predicted with NN or N are treated as

[6] 345 samples with both positive and negative labels are treated as neutral.

disagreement; the ones predicted as PP or P are agreement; O is neutral.

Comparison

We compare with two baselines. (1) **Baseline (Polarity)** is based on counting the sentiment words from our lexicon. An utterance or segment is predicted as agreement if it contains more positive words than negative words, or disagreement if more negative words are observed. Otherwise, it is neutral. (2) **Baseline (Distance)** is extended from Hassan et al. [2010]. Each sentiment word is associated with the closest second person pronoun, and a surface distance can be computed between them. A classifier based on Support Vector Machines [Joachims, 1999] (SVM) is trained with the features of sentiment words, minimum/maximum/average of the distances.

We also compare with two state-of-the-art methods that are widely used in sentiment prediction for conversations. The first one is an RBF kernel SVM based approach, which has been used for sentiment prediction [Hassan et al., 2010], and (dis)agreement detection [Yin et al., 2012] in online debates. The second is linear chain CRF, which has been utilized for (dis)agreement identification in broadcast conversations [Wang et al., 2011].

8.2.4 Results

In this section, we first show the experimental results on sentence- and segment-level agreement and disagreement detection in two types of online discussions – *Wikipedia Talk pages* and *online debates*. Then we provide more detailed analysis for the features used in our model. Furthermore, we discuss several types of errors made in the model.

Wikipedia Talk Pages

We evaluate the systems by standard F1 score on each of the three categories: agreement, disagreement, and neutral. For AAWD, we compute two versions of F1 scores. **Strict F1** is computed against the true labels. For **soft F1**, if a sentence is never labeled by any annotator on the sentence-level and adopts its agreement/disagreement label from the turn-level annotation, then it is treated as a true positive when predicted as neutral.

Table 8.3 demonstrates our main results on the Wikipedia Talk pages (AAWD dataset). Without downsampling, our isotonic CRF based systems with the new lexicon significantly outperform the compared approaches for agreement and disagreement detection according to the paired-t test ($p < 0.05$). We also perform downsampling by removing the turns only containing neutral utterances. However, it does not always help with performance. We suspect that, with less neutral samples in the training data, the classifier is less likely to make neutral predictions, which thus decreases true positive predictions. For strict F-scores on agreement/disagreement, downsampling has mixed effect, but mostly we get slightly better performance.

Online Debates

Similarly, F1 scores for agreement, disagreement and neutral for online debates (IAC dataset) are displayed in Table 8.4. Both of our systems based on isotonic CRF achieve significantly better F1 scores than the comparison. Especially, our system with the new lexicon produces the best results. For SVM and linear-chain CRF based systems, we also add new sentiment features constructed from the new lexicon as described in Section 8.2.1. We can see that those sentiment features also boost the performance for both of the compared approaches.

	Strict F1			Soft F1		
	Agree	**Disagree**	**Neutral**	**Agree**	**Disagree**	**Neutral**
Baseline (Polarity)	14.56	25.70	64.04	22.53	38.61	66.45
Baseline (Distance)	8.08	20.68	84.87	33.75	55.79	88.97
SVM (3-way)	26.76	35.79	77.39	44.62	52.56	80.84
+ downsampling	21.60	36.32	72.11	31.86	49.58	74.92
CRF (3-way)	20.99	23.85	85.28	56.28	56.37	89.41
CRF (5-way)	20.47	19.42	85.86	58.39	56.30	90.10
+ downsampling	24.26	31.28	77.12	47.30	46.24	80.18
isotonic CRF	24.32	21.95	86.26	68.18	62.53	88.87
+ downsampling	29.62	34.17	80.97	55.38	53.00	84.56
+ new lexicon	**46.01**	***51.49***	*87.40*	***74.47***	***67.02***	*90.56*
+ new lexicon + downsampling	***47.90***	**49.61**	81.60	64.97	58.97	84.04

Table 8.3: Strict and soft F1 scores for agreement and disagreement detection on Wikipedia talk pages (AAWD). All the numbers are multiplied by 100. In each column, **bold** entries (if any) are statistically significantly higher than all the rest, and the *italic* entry has the highest absolute value. Our model based on the isotonic CRF with the new lexicon produces significantly better results than all the other systems for agreement and disagreement detection. Downsampling, however, is not always helpful.

Feature Evaluation

Moreover, we evaluate the effectiveness of features by adding one type of features each time. The results are listed in Table 8.5. As it can be seen, the performance gets improved incrementally with every new set of features.

We also utilize χ^2-test to highlight some of the salient features on the two datasets. We can see from Table 8.6 that, for online debates (IAC), some features are highly topic related, such as "*the male*" or "*the scientist*". This observation concurs with the conclusion in Misra and Walker [2013] that features with topic information are indicative for agreement and disagreement detection.

	Agree	Disagree	Neu
Baseline (Polarity)	3.33	5.96	65.61
Baseline (Distance)	1.65	5.07	85.41
SVM (3-way)	25.62	69.10	31.47
+ new lexicon features	28.35	72.58	34.53
CRF (3-way)	29.46	74.81	31.93
CRF (5-way)	24.54	69.31	39.60
+ new lexicon features	28.85	71.81	39.14
isotonic CRF	**53.40**	**76.77**	*44.10*
+ new lexicon	***61.49***	***77.80***	*51.43*

Table 8.4: F1 scores for agreement and disagreement detection on online debate (IAC). All the numbers are multiplied by 100. In each column, **bold** entries (if any) are statistically significantly higher than all the rest, and the *italic* entry has the highest absolute value except baselines. We have two main observations: 1) Both of our models based on isotonic CRF significantly outperform other systems for agreement and disagreement detection. 2) By adding the new lexicon, either as features or constraints in isotonic CRF, all systems achieve better F1 scores.

Error Analysis

After a closer look at the data, we found two major types of errors. Firstly, people express disagreement not only by using opinionated words, but also by providing contradictory example. This needs a deeper understanding of the semantic information embedded in the text. Techniques like textual entailment can be used in the further work. Secondly, a sequence of sentences with sarcasm is hard to detect. For instance, *"Bravo, my friends! Bravo! Goebbles would be proud of your abilities to whitewash information."* We observe terms like "Bravo", "friends", and "be proud of" that are indicators for positive sentiment; however, they are in sarcastic tone. We believe a model that is able to detect sarcasm would further improve the performance.

AAWD	Agree	Disagree	Neu
Lex	40.77	52.90	79.65
Lex + Syn	68.18	63.91	88.87
Lex + Syn + Disc	70.93	63.69	89.32
Lex + Syn + Disc + Con	71.27	63.72	89.60
Lex + Syn + Disc + Con + Sent	**74.47**	**67.02**	90.56

IAC	Agree	Disagree	Neu
Lex	56.65	75.35	45.72
Lex + Syn	54.16	75.13	46.12
Lex + Syn + Disc	54.27	76.41	47.60
Lex + Syn + Disc + Con	55.31	77.25	48.87
Lex + Syn + Disc + Con + Sent	**61.49**	77.80	**51.43**

Table 8.5: Results on Wikipedia talk page (AAWD) (with soft F1 score) and online debate (IAC) with different feature sets (i.e **Lex**ical, **Syn**tacitc/Semantic, **Disc**ourse, **Con**versation, and **Sent**iment features) by using isotonic CRF. The numbers in **bold** are statistically significantly higher than the numbers above it (paired-t test, $p < 0.05$).

AAWD
POSITIVE: agree, nsubj (agree, I), nsubj (right, you), Rel (Sentiment$_{pos}$, I), thanks, amod (idea, good), nsubj(glad, I), good point, concur, happy with, advmod (good, pretty), suggestion$_{Hedge}$
NEGATIVE: you, your, nsubj (negative, you), numberOfNegator, don't, nsubj (disagree, I), actually$_{SentInitial}$, please stop$_{SentInitial}$, what ?$_{SentInitial}$, should$_{Hedge}$
IAC
POSITIVE: amod (conclusion, logical), Rel (agree, on), Rel (have, justified), Rel (work, out), one might$_{SentInitial}$, to confirm$_{Hedge}$, women
NEGATIVE: their kind, the male, the female, the scientist, according to, is stated, poss (understanding, my), hell$_{SentInitial}$, whatever$_{SentInitial}$

Table 8.6: Relevant features by χ^2 test on AAWD and IAC datasets.

8.3 Online Dispute Detection with Sentiment Analysis Approach

8.3.1 Data Construction: A Dispute Corpus

We construct the first dispute detection corpus to date; it consists of dispute and non-dispute discussions from Wikipedia Talk pages.

Step 1: Get Talk Pages of Disputed Articles. Wikipedia articles are edited by different editors. If an article is observed to have disputes on its *talk page*, editors can assign dispute tags to the article to flag it for attention. In this research, we are interested in talk pages whose corresponding articles are labeled with the following tags: DISPUTED, TOTALLYDISPUTED, DISPUTED-SECTION, TOTALLYDISPUTED-SECTION, POV. The tags indicate that an article is disputed, or the neutrality of the article is disputed (POV).

We use the 2013-03-04 Wikipedia data dump, and extract talk pages for articles that are labeled with dispute tags by checking the revision history. This results in 19,071 talk pages.

Step 2: Get Discussions with Disputes. Dispute tags can also be added to *talk pages* themselves. Therefore, in addition to the tags mentioned above, we also consider the "Request for Comment" (RFC) tag on talk pages. According to Wikipedia[7], RFC is used to request outside opinions concerning the disputes.

3609 discussions are collected with dispute tags found in the revision history. We further classify dispute discussions into three subcategories: CONTROVERSY, REQUEST FOR COMMENT (RFC), and RESOLVED based on the tags found in discussions (see Table 8.7). The numbers of discussions for the three types are 42, 3484, and 105, respectively. Note that dispute tags only appear in a normalsize number of articles and talk pages. There may exist other discussions with disputes.

Dispute Subcategory	**Wikipedia Tags on Talk pages**
Controversy	CONTROVERSIAL, TOTALLYDISPUTED, DISPUTED, CALM TALK, POV
Request for Comment	RFC
Resolved	Any tag from above + RESOLVED

Table 8.7: Subcategory for disputes with corresponding tags. Note that each discussion in the RESOLVED class has more than one tag.

[7] http://en.wikipedia.org/wiki/Wikipedia:Requests_for_comment

Step 3: Get Discussions without Disputes. Likewise, we collect non-dispute discussions from pages that are never tagged with disputes. We consider non-dispute discussions with at least 3 distinct speakers and 10 turns. 3609 discussions are randomly selected with this criterion. The average turn numbers for dispute and non-dispute discussions are 45.03 and 22.95, respectively.

8.3.2 Sentence-level Sentiment Prediction

This section describes our sentence-level sentiment tagger, from which we construct features for dispute detection (Section 8.3.3).

Consider a discussion comprised of sequential turns; each turn consists of a sequence of sentences. Our model takes as input the sentences $\mathbf{x} = \{x_1, \cdots, x_n\}$ from a single turn, and outputs the corresponding sequence of sentiment labels $\mathbf{y} = \{y_1, \cdots, y_n\}$, where $y_i \in \mathcal{O}, \mathcal{O} = \{\text{NN}, \text{N}, \text{O}, \text{P}, \text{PP}\}$. The labels in $\mathcal{O}$ represent very negative (NN), negative (N), neutral (O), positive (P), and very positive (PP), respectively.

Given that traditional Conditional Random Fields (CRFs) [Lafferty et al., 2001a] ignore the ordinal relations among sentiment labels, we choose *isotonic CRFs* [Mao and Lebanon, 2007] for sentence-level sentiment analysis as they can enforce monotonicity constraints on the parameters consistent with the ordinal structure and domain knowledge (e.g. word-level sentiment conveyed via a lexicon). Concretely, we take a lexicon $\mathcal{M} = \mathcal{M}_p \cup \mathcal{M}_n$, where $\mathcal{M}_p$ and $\mathcal{M}_n$ are two sets of features (usually words) identified as strongly associated with positive and negative sentiment. Assume $\mu_{\langle\sigma,w\rangle}$ encodes the weight between label σ and feature w, for each feature $w \in \mathcal{M}_p$; then the isotonic CRF enforces $\sigma \leq \sigma' \Rightarrow \mu_{\langle\sigma,w\rangle} \leq \mu_{\langle\sigma',w\rangle}$. For example, when we observe "totally agree" in the training data, the feature parameter for $\mu_{\langle\text{PP,totally agree}\rangle}$ is likely to increase. Similar

Lexical Features	**Syntactic/Semantic Features**
- unigram/bigram	- unigram with POS tag
- number of words all uppercased	- dependency relation
- number of words	**Conversation Features**
Discourse Features	- quote overlap with target
- initial uni-/bi-/tri-gram	- TFIDF similarity with target
- repeated punctuations	(remove quote first)
- hedging phrases collected from	**Sentiment Features**
Farkas et al. [2010]	- connective + sentiment words
- number of negators	- sentiment dependency relation
	- sentiment words

Table 8.8: Features used in sentence-level sentiment prediction. Numerical features are first normalized by standardization, then binned into 5 categories.

constraints are defined on $\mathcal{M}_n$.

Our lexicon is built by combining MPQA [Wilson et al., 2005], General Inquirer [Stone et al., 1966], and SentiWordNet [Esuli and Sebastiani, 2006] lexicons. Words with contradictory sentiments are removed. We use the features in Table 8.8 for sentiment prediction.

Syntactic/Semantic Features. We have two versions of dependency relation features, the original form and a form that generalizes a word to its POS tag, e.g. "nsubj(wrong, you)" is generalized to "nsubj(`ADJ`, you)" and "nsubj(wrong, `PRP`)".

Discourse Features. We extract the initial unigram, bigram, and trigram of each utterance as discourse features [Hirschberg and Litman, 1993].

Conversation Features. Conversation features encode some useful information regarding the similarity between the current utterance(s) and the sentences uttered by the target participant. TFIDF similarity is computed. We also check if the current utterance(s) quotes target sentences and compute its length.

Sentiment Features. In addition to items in the sentiment lexicon, We gather connec-

tives from the Penn Discourse TreeBank [Prasad et al., 2008] and combine them with any sentiment word that precedes or follows it as new features. Sentiment dependency relations are the dependency relations that include a sentiment word. We replace those words with their polarity equivalents. For example, relation "nsubj(wrong, you)" becomes "nsubj($\texttt{SentiWord}_{neg}$, you)".

8.3.3 Online Dispute Detection

In this section, we investigate whether we can leverage the sentiment tagging model to the task of *Online Dispute Detection*.

Training A Sentiment Classifier

Dataset. We train the sentiment classifier using the *Authority and Alignment in Wikipedia Discussions (AAWD)* corpus [Bender et al., 2011] on a 5-point scale (i.e. NN, N, O, P, PP). AAWD consists of 221 English Wikipedia discussions with positive and negative alignment annotations.[8] The average turn number is 15.77 for each discussion.

Annotators either label each sentence as positive, negative or neutral, or label the full turn. For instances that have only a turn-level label, we assume all sentences have the same label as the turn. We further transform the labels into the five sentiment labels. Sentences annotated as being a positive alignment by at least two annotators are treated as very positive (PP). If a sentence is only selected as positive by one annotator or obtains the label via turn-level annotation, it is positive (P). Very negative (NN) and negative (N) are collected in the same way. All others are neutral (O). Among all 16,501 sentences

[8] Bender et al. [2011] originally use positive alignment and negative alignment to indicate two types of social moves. They define those alignment moves as "agreeing or disagreeing" with the target. We thus use agreement and disagreement instead of positive and negative alignment in this work.

in AAWD, 1,930 and 1,102 are labeled as NN and N. 532 and 99 of them are PP and P. The other 12,648 are considered neutral.

Evaluation. To evaluate the performance of the sentiment tagger, we compare to two baselines. (1) **Baseline (Polarity)**: a sentence is predicted as positive if it has more positive words than negative words, or negative if more negative words are observed. Otherwise, it is neutral. (2) **Baseline (Distance)** is extended from Hassan et al. [2010]. Each sentiment word is associated with the closest second person pronoun, and a surface distance is computed. An SVM classifier [Joachims, 1999] is trained using features of the sentiment words and minimum/maximum/average of the distances.

We also compare with two state-of-the-art methods that are used in sentiment prediction for conversations: (1) an SVM (RBF kernel) that is employed for identifying sentiment-bearing sentences [Hassan et al., 2010], and (dis)agreement detection [Yin et al., 2012] in online debates; (2) a Linear CRF for (dis)agreement identification in broadcast conversations [Wang et al., 2011].

We evaluate the systems using standard F1 on classes of positive, negative, and neutral, where samples predicted as PP and P are positive alignment, and samples tagged as NN and N are negative alignment. Table 8.9 describes the main results on the AAWD dataset: our isotonic CRF based system significantly outperforms the alternatives for positive and negative alignment detection (paired-t test, $p < 0.05$).

Dispute Detection

We model dispute detection as a standard binary classification task, and investigate four major types of features as described below.

	Pos	Neg	Neutral
Baseline (Polarity)	22.53	38.61	66.45
Baseline (Distance)	33.75	55.79	88.97
SVM (3-way)	44.62	52.56	80.84
CRF (3-way)	56.28	56.37	89.41
CRF (5-way)	58.39	56.30	90.10
isotonic CRF	**68.18**	**62.53**	88.87

Table 8.9: F1 scores for positive and negative alignment on Wikipedia Talk pages (AAWD) using 5-fold cross-validation. In each column, **bold** entries (if any) are statistically significantly higher than all the rest. We also compare with an SVM and linear CRF trained with three classes (3-way). Our model based on the isotonic CRF produces significantly better results than all the other systems.

Lexical Features. We first collect `unigram` and `bigram` features for each discussion.

Topic Features. Articles on specific topics, such as politics or religions, tend to arouse more disputes. We thus extract the `category` information of the corresponding article for each talk page. We further utilize `unigrams` and `bigrams` of the category as topic features.

Discussion Features. This type of feature aims to capture the structure of the discussion. Intuitively, the more turns or the more participants a discussion has, the more likely there is a dispute. Meanwhile, participants tend to produce longer utterances when they make arguments. We choose `number of turns`, `number of participants`, `average number of words in each turn` as features. In addition, the frequency of revisions made during the discussion has been shown to be good indicator for controversial articles [Vuong et al., 2008], that are presumably prone to have disputes. Therefore, we encode the `number of revisions` that happened during the discussion as a feature.

Sentiment Features. This set of features encode the sentiment distribution and transition in the discussion. We train our sentiment tagging model on the full AAWD dataset,

and run it on the Wikipedia dispute corpus.

Given that consistent negative sentiment flow usually indicates an ongoing dispute, we first extract features from `sentiment distribution` in the form of `number/probability of sentiment per type`. We also estimate the `sentiment transition` probability $P(S_t \rightarrow S_{t+1})$ from our predictions, where S_t and S_{t+1} are sentiment labels for the current sentence and the next. We then have features as `number/portion of sentiment transitions per type`.

Features described above mostly depict the *global* sentiment flow in the discussions. We further construct a *local* version of them, since sentiment distribution may change as discussion proceeds. For example, less positive sentiment can be observed as dispute being escalated. We thus split each discussion into three equal length stages, and create sentiment distribution and transition features for each stage.

Results and Error Analysis. We experiment with logistic regression, SVM with linear and RBF kernels, which are effective methods in multiple text categorization tasks [Joachims, 1999, Zhang and J. Oles, 2001]. We normalize the features by standardization and conduct a 5-fold cross-validation. Two baselines are listed: (1) labels are randomly assigned; (2) all discussions have disputes.

Main results for different classifiers are displayed in Table 8.11. All learning based methods outperform the two baselines, and among them, SVM with the RBF kernel achieves the best F1 score and accuracy (0.78 and 0.80). Experimental results with various combinations of features sets are displayed in Table 8.11. As it can be seen, sentiment features obtains the best accuracy among the four types of features. A combination of topic, discussion, and sentiment features achieves the best performance on recall, F1, and accuracy. Specifically, the accuracy is significantly higher than all the

other systems (paired-t test, $p < 0.05$).

	Prec	**Rec**	**F1**	**Acc**
Baseline (Random)	50.00	50.00	50.00	50.00
Baseline (All dispute)	50.00	100.00	66.67	50.00
Logistic Regression	74.76	72.29	73.50	73.94
SVM_{Linear}	69.81	71.90	70.84	70.41
SVM_{RBF}	**77.38**	79.14	**78.25**	**80.00**

Table 8.10: Dispute detection results on Wikipedia Talk pages. The numbers are multiplied by 100. The items in **bold** are statistically significantly higher than others in the same column (paired-t test, $p < 0.05$). SVM with the RBF kernel achieves the best performance in precision, F1, and accuracy.

	Prec	**Rec**	**F1**	**Acc**
Lexical (Lex)	75.86	34.66	47.58	61.82
Topic (Top)	68.44	71.46	69.92	69.26
Discussion (Dis)	69.73	76.14	72.79	71.54
Sentiment (Senti_{g+l})	72.54	69.52	71.00	71.60
Top + Dis	68.49	71.79	70.10	69.38
Top + Dis + Senti_{g}	77.39	78.36	77.87	77.74
Top + Dis + Senti_{g+l}	77.38	*79.14*	*78.25*	**80.00**
Lex + Top + Dis + Senti_{g+l}	*78.38*	75.12	76.71	77.20

Table 8.11: Dispute detection results with different feature sets by SVM with RBF kernel. The numbers are multiplied by 100. Senti_{g} represents global sentiment features, and Senti_{g+l} includes both global and local features. The number in **bold** is statistically significantly higher than other numbers in the same column (paired-t test, $p < 0.05$), and the *italic* entry has the highest absolute value.

After a closer look at the results, we find two main reasons for incorrect predictions. Firstly, errors from sentiment prediction get propagated into dispute detection. Due to the limitation of existing general-purpose lexicons, some opinionated dialog-specific terms are hard to catch. For example, "I told you over and over again..." strongly suggests a negative sentiment, but no single word shows negative connotation. Constructing a lexicon tuned for conversational text might further improve the performance. Secondly, some dispute discussions are harder to detect than the others due to different dialog structures. For instance, the recalls for dispute discussions of "controversy", "RFC", and "resolved" are 0.78, 0.79, and 0.86 respectively. We intend to design mod-

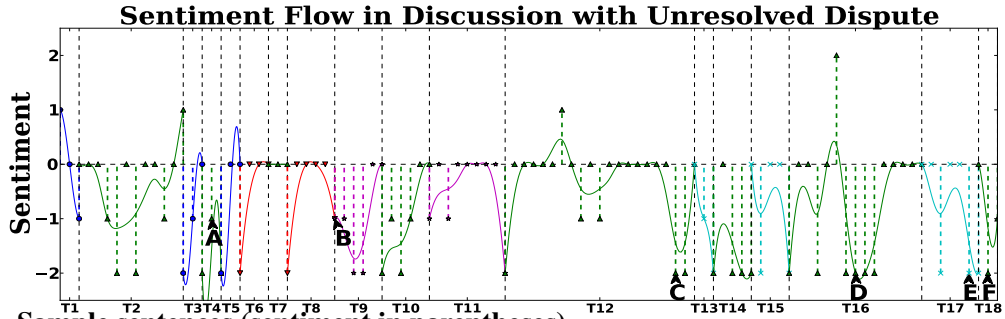
els that are able to capture dialog structures, such as pragmatic information, in the future work.

Sentiment Flow Visualization. We visualize the sentiment flow of two disputed discussions in Figure 8.3. The plots reveal persistent negative sentiment in unresolved disputes (top). For the resolved dispute (bottom), participants show gratitude when the problem is settled.

8.4 Conclusion

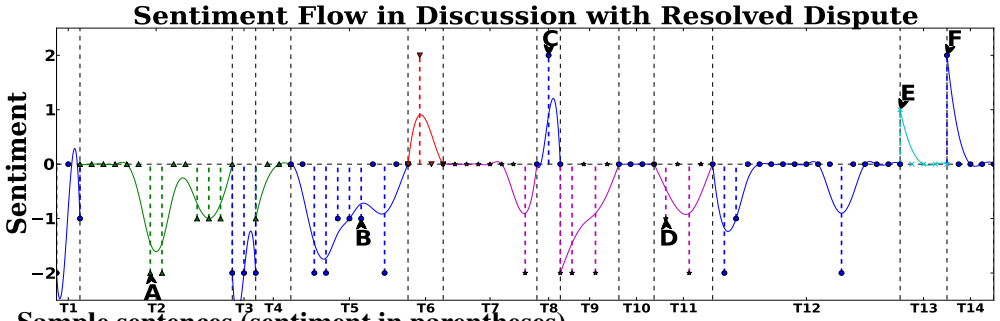
In this chapter, we first presented an agreement and disagreement detection model based on isotonic CRFs that outputs labels at the sentence- or segment-level. We bootstrapped the construction of a sentiment lexicon for online discussions, encoding it in the form of domain knowledge for the isotonic CRF learner. Our sentiment-tagging model was shown to outperform the state-of-the-art approaches on both Wikipedia Talk pages and online debates.

We then presented a sentiment analysis-based approach to online dispute detection. We created a large-scale dispute corpus from Wikipedia Talk pages to study the problem. A sentiment prediction model based on isotonic CRFs was proposed to output sentiment labels at the sentence-level. Experiments on our dispute corpus also demonstrated that classifiers trained with sentiment tagging features outperformed others that do not.



Sample sentences (sentiment in parentheses)

A: no, I sincerely plead with you... (N) If not, you are just wasting my time. (NN)
B: I believe Sweet's proposal... is quite silly. (NN)
C: Tell you what. (NN) If you can get two other editors to agree... I will shut up and sit down. (NN)
D: But some idiot forging your signature claimed that doing so would violate. (NN)... Please go have some morning coffee. (O)
E: And I don't like coffee. (NN) Good luck to you. (NN)
F: Was that all? (NN)... I think that you are in error... (N)



Sample sentences (sentiment in parentheses)

A: So far so confusing. (NN)...
B: ... I can not see a rationale for the landrace having its own article... (N) With Turkish Van being a miserable stub, there's no such rationale for forking off a new article... (NN)...
C: I've also copied your post immediately above to that article's talk page since it is a great "nutshell" summary. (PP)
D: Err.. how can the opposite be true... (N)
E: Thanks for this, though I have to say some of the facts floating around this discussion are wrong. (P)
F: Great. (PP) Let's make sure the article is clear on this. (O)

Figure 8.3: Sentiment flow for a discussion with **unresolved** dispute about the definition of "white people" (top) and a discussion with **resolved** dispute on merging articles about van cat (bottom). The labels $\{\text{NN}, \text{N}, \text{O}, \text{P}, \text{PP}\}$ are mapped to $\{-2, -1, 0, 1, 2\}$ in sequence. Sentiment values are convolved by using a Gaussian smoothing kernel, and then cubic-spline interpolation is conducted. Different speakers are represented by curves of different colors. Dashed vertical lines delimit turns. Representative sentences are labeled with letters and their sentiment labels are shown on the right. For unresolved dispute (top), we see that negative sentiment exists throughout the discussion. Whereas, for the resolved dispute (bottom), less negative sentiment is observed at the end of the discussion; participants also show appreciation after the problem is solved (e.g. E and F in the plot).

CHAPTER 9

CONCLUSION AND FUTURE HORIZONS

9.1 Conclusion

During the past decades, we have witnessed the amount of information on the Internet growing explosively. Information overload has thus become an inevitable challenge for almost every application domain that requires organizing, summarizing, searching, or filtering large amounts of digital data. Meanwhile, we are dealing with heterogeneous textual data of disparate types and genres, ranging from highly edited documents, such as news articles or editorials, to informal text in social media with noisy information on different levels, such as restaurant reviews or microblogs. More importantly, textual data derived from social media is difficult to analyze using existing NLP tools that are designed for processing edited text.

Another important factor that needs to be considered for designing NLP systems is the users' information need, which could be interpreted broadly. One of the goals for developing NLP techniques is to facilitate knowledge learning from textual data. It is natural to add user modeling as an important component. However, users' information needs can be diverse and they may change over time. Moreover, the disparity of users' knowledge levels would require information to be presented in different ways.

Motivated by the above challenges, this dissertation has proposed general-purpose natural language processing techniques that efficiently analyze textual data from diverse domains and different genres. Specifically, we make progress in the following two areas: (1) generating high quality summaries to satisfy users' information request, and (2) studying the sentiments and opinions expressed from the online conversations to better

understand social interactions. Although the approaches presented here by no means fully address all these challenges, they show promising research directions for understanding large amounts of socially-generated textual data. Particularly, we tackle the problem of analyzing textual data with inherent noise to meet users' information need. We show the effectiveness of summarization and sentiment analysis techniques, which will motivate future research in these areas.

Concretely, this dissertation presents contributions in text summarization and sentiment analysis for the analyze of large amounts of socially-generated textual content. For text summarization, we propose domain-independent abstract generation frameworks for focused meeting summarization in Chapters 3 and 4. We show that abstractive summarization methods are capable of extracting salient information and presenting it in a human comprehensive way. Meanwhile, they are also powerful at removing redundant and noisy content from the input textual data.

Furthermore, we tackle the challenge of constructing summaries to address users' queries that are in the form of open-ended questions. The sentence compression framework described in Chapter 5 demonstrates how to remove auxiliary information from lengthy sentences while preserving relevant information that users ask for. The opinion summarization framework presented in Chapter 6 provides a way to collect opinions of high diversity from online social media and present it as text summaries.

Finally, we present a socially-informed timeline generation system in Chapter 7. Our system generates a news article summary and a user comment summary on a daily basis for an ongoing complex event, while existing work only considers summarizing one type of data. Our work describes an effective approach to build the connection between the events reported by traditional news media and the relevant public opinions on social media. We show that the news articles and user comments can provide complemen-

tary information, and the usage of two sources of information can boost users' reading experience and help them absorb information in an efficient way.

This dissertation also makes steps towards deeper understanding of online social dynamics. Sentiment analysis methods are designed to study how people interact in the settings of online collaboration or online debate. In Chapter 8, we first present a sequential model to identify agreement and disagreement in online discussion on sentence- or segment-level. Traditional sentiment lexicons are usually constructed from news articles, and thus have limited coverage for sentiment words used in social media. We therefore automatically construct a socially-tuned sentiment lexicon from millions of online discussions. This sentiment classifier is also used to investigate the task of online dispute detection. We then collect the first online dispute detection dataset, and construct classifiers with different types of features to predict dispute/non-dispute label for a discussion.

9.2 Future Horizons

In this final section, we discuss the potential future directions for text summarization and NLP techniques for computational social science. We also provide visions on how to apply the techniques proposed in this dissertation and their extensions to boost interdisciplinary research.

There is still a long way to go for generating high quality abstracts for open domain documents. As Cheung and Penn [2013] point out, domain inference by making use of in-domain knowledge sources could advance abstraction. Meanwhile, the linguistic quality of the current automatically generated summaries is still far from satisfying. To address those challenges, we need advanced text generation systems that are able

to improve the coherence, clarity, and conciseness. Given that abstract generation is still a nascent field, there is an abundance of exciting problems to solve. For instance, the research community has invested substantial effort on generating grammatical and informative summaries, but has really ignored the fact that users may prefer summaries personalized according to their interests and literacy level. It is more desirable to build language generation systems that can produce customized summaries for different types of readers and disparate genres of data.

Another research problem intrinsic to language generation is the issue of how to more broadly evaluate the quality of the generated text: whether the text is well-organized and persuasive, how easy-to-read the text is, and whether the text is entertaining to read. To achieve this goal, we need high-level NLP tools for discourse and argumentation analysis, metaphor understanding, coherence and cohesion modeling, etc. The techniques developed can also be used for educational purposes, for example, essay scoring. Another interesting direction for building summarization system is to develop interactive algorithms that can leverage human-computer interaction techniques to extract instant user feedbacks for incorporation into the summarization system. With user's feedback as guidance, the system is capable of adjusting the output in real time, thus enabling human guidance in the summarization process.

Our work on the usage of summarization, sentiment analysis, and information extraction techniques in conversation modeling has demonstrated the potential of leveraging conversational data for knowledge discovery. One future direction is to link the findings on personal interactions to computational social science. For example, online debate forums provide a place for people to discuss and argue on complex issues of social significance. In addition to summarizing the main arguments people present and the corresponding supporting materials, it would be good to develop computational models

to identify the effective debating strategies and to determine how the choice of language affects public opinion.

With the emergence of massive amounts of online and offline textual data, NLP methods can be used as enabling tools to help people understand and absorb knowledge in different domains, especially the ones they are not familiar with. For instance, applying NLP approaches to medical domain is a rising research field. It is also very challenging because the presence of specialized vocabulary requires extra knowledge to fully understand medical documents. We can develop domain-specific text summarization and information extraction techniques that can take into consideration an end user's level of medical literacy. Such techniques will benefit patients, physicians, and researchers, and thus lead to a real impact on society.

Finally, evaluating the quality of writing can be a time-consuming task, especially when the number of documents is large. Some compelling examples include massive open online courses (MOOC) in writing as well as language testing services. Furthermore, it is non-trivial for language assessment algorithms to detect the logical flow and argumentation structure in the text. It would be helpful to apply NLP techniques to tackle such challenges and facilitate the automatic grading of essays using the aforementioned linguistic quality evaluation models [Burstein, 2003, Tetreault and Chodorow, 2008]. This will pave the way for building more effective educational tools, as it will not only reduce the educators' burden of grading, but also provide students with instant personalized advice on writing skills.

BIBLIOGRAPHY

Rob Abbott, Marilyn Walker, Pranav Anand, Jean E. Fox Tree, Robeson Bowmani, and Joseph King. How can you say such things?!?: Recognizing disagreement in informal political argument. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 2–11, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-96-1.

Amjad Abu-Jbara, Mona Diab, Pradeep Dasigi, and Dragomir Radev. Subgroup detection in ideological discussions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 399–409, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics.

Alfred V. Aho and Jeffrey D. Ullman. Syntax directed translations and the pushdown assembler. *J. Comput. Syst. Sci.*, 3(1):37–56, 1969.

J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, Lansdowne, VA, USA, February 1998. 007.

James Allan, Rahul Gupta, and Vikas Khandelwal. Temporal summaries of new topics. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research*

and Development in Information Retrieval, SIGIR '01, pages 10–18, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6. doi: 10.1145/383952.383954.

Kelsey Allen, Giuseppe Carenini, and Raymond Ng. Detecting disagreement in conversations using pseudo-monologic rhetorical structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1169–1180, Doha, Qatar, October 2014. Association for Computational Linguistics.

Miguel Almeida and Andre Martins. Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 196–206, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Hadi Amiri, Zheng-Jun Zha, and Tat-Seng Chua. A pattern matching based model for implicit opinion question identification. In *AAAI*. AAAI Press, 2013. ISBN 978-1-57735-615-8.

Gabor Angeli, Percy Liang, and Dan Klein. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 502–512, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. Multi-document abstractive summarization using ilp based multi-sentence compression. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI '15, 2015.

Michele Banko and Oren Etzioni. The Tradeoffs Between Open and Traditional Relation Extraction. In *Proceedings of ACL-08: HLT*, Columbus, Ohio, 2008.

Michele Banko, Michael J Cafarella, Stephen Soderl, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *In IJCAI*, pages 2670–2676, 2007.

Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 141–148, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219858.

Regina Barzilay and Lillian Lee. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 16–23, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073448.

Regina Barzilay and Kathleen R. McKeown. Sentence fusion for multidocument news summarization. *Comput. Linguist.*, 31(3):297–328, September 2005. ISSN 0891-2017. doi: 10.1162/089120105774321091.

Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. Inferring strategies for sentence ordering in multidocument news summarization. *J. Artif. Int. Res.*, 17(1): 35–55, August 2002. ISSN 1076-9757.

Tal Baumel, Raphael Cohen, and Michael Elhadad. Query-chain focused summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 913–922, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

Emily M. Bender, Jonathan T. Morgan, Meghan Oxley, Mark Zachry, Brian Hutchinson, Alex Marin, Bin Zhang, and Mari Ostendorf. Annotating social acts: Authority claims

and alignment moves in wikipedia talk pages. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 48–57, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-96-1.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. Jointly learning to extract and compress. ACL '11, pages 481–490, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, March 1996. ISSN 0891-2017.

H Bhandari, T Ito, M Shimbo, and Y Matsumoto. Generic text summarization using probabilistic latent semantic indexing. In *Proceedings of IJCNLP*, pages 133–140, 2008.

Matt Billings and Leon Adam Watts. Understanding dispute resolution online: using text to reflect personal and substantive issues in conflict. In Elizabeth D. Mynatt, Don Schoner, Geraldine Fitzpatrick, Scott E. Hudson, W. Keith Edwards, and Tom Rodden, editors, *CHI*, pages 1447–1456. ACM, 2010. ISBN 978-1-60558-929-9.

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, 2009.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435.

Eric Breck, Yejin Choi, and Claire Cardie. Identifying expressions of opinion in context. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-2007)*, Hyderabad, India, January 2007.

T. Briscoe and J. Carroll. Robust accurate statistical annotation of general text, 2002.

Samuel Brody and Noemie Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 804–812, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5.

Trung H. Bui, Matthew Frampton, John Dowding, and Stanley Peters. Extracting decisions from multi-party dialogue using directed graphical models and semantic similarity. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '09, pages 235–243, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-64-0.

Christopher J. C. Burges. From RankNet to LambdaRank to LambdaMART: An Overview. Technical report, Microsoft Research, 2010.

Christopher J.C. Burges, Robert Ragno, and Quoc Viet Le. Learning to rank with nonsmooth cost functions. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 193–200. MIT Press, Cambridge, MA, 2007.

Jill Burstein. The e-rater® scoring engine: Automated essay scoring with natural language processing. 2003.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 129–136, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273513.

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 AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2153–2159, 2015.

Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM. ISBN 1-58113-015-5. doi: 10.1145/290941.291025.

Giuseppe Carenini, Gabriel Murray, and Raymond Ng. *Methods for Mining and Summarizing Text Conversations*. Morgan & Claypool Publishers, 2011.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, and Mccowan Wilfried Post Dennis Reidsma. The ami meeting corpus: A pre-announcement. In *In Proc. MLMI*, pages 28–39, 2005.

Asli Celikyilmaz and Dilek Hakkani-Tür. Discovery of topically coherent sentences for extractive summarization. ACL '11, pages 491–499, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9.

Asli Celikyilmaz, Dilek Hakkani-Tur, and Gokhan Tur. LDA Based Similarity Modeling for Question Answering. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, SS '10, pages 1–9, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Barun Chandra and Magnús M. Halldórsson. Facility dispersion and remote subgraphs. In *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory*, SWAT '96, pages 53–65, London, UK, UK, 1996. Springer-Verlag. ISBN 3-540-61422-2.

Angel X. Chang and Christopher Manning. Sutime: A library for recognizing and normalizing time expressions. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uur Doan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Jonathan Chang, Jordan Boyd-Graber, and David M. Blei. Connections between the lines: Augmenting social networks with text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 169–178, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557044.

Berlin Chen and Yi-Ting Chen. Extractive spoken document summarization for information retrieval. *Pattern Recogn. Lett.*, 29:426–437, March 2008. ISSN 0167-8655. doi: http://dx.doi.org/10.1016/j.patrec.2007.10.022.

Harr Chen, Edward Benson, Tahira Naseem, and Regina Barzilay. In-domain relation discovery with meta-constraints via posterior regularization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 530–540, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

Jackie Chi Kit Cheung and Gerald Penn. Towards robust abstractive multi-document summarization: A caseframe analysis of centrality and domain. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1233–1242, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Jackie Chi Kit Cheung and Gerald Penn. Unsupervised sentence enhancement for automatic summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 775–786, Doha, Qatar, October 2014. Association for Computational Linguistics.

Hai Leong Chieu and Yoong Keok Lee. Query based event extraction along a timeline. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 425–432, New York, NY, USA, 2004. ACM. ISBN 1-58113-881-4. doi: 10.1145/1008992.1009065.

Yejin Choi and Claire Cardie. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 590–598, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-62-6.

Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-2005)*, pages 355–362, Vancouver, Canada, 2005.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. Towards coherent multi-document summarization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1173, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

Janara Christensen, Stephen Soderland, Gagan Bansal, and Mausam. Hierarchical summarization: Scaling up multi-document summarization. In *Proceedings of the 52nd*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 902–912, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

Claudio Cioffi-Revilla. *Introduction to Computational Social Science: Principles and Applications*. Springer Publishing Company, Incorporated, 2014. ISBN 1447156609, 9781447156604.

James Clarke and Mirella Lapata. Global inference for sentence compression an integer linear programming approach. *J. Artif. Int. Res.*, 31(1):399–429, March 2008. ISSN 1076-9757.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 152–159, Stroudsburg, PA, USA, 2006a. Association for Computational Linguistics.

John M. Conroy, Judith D. Schlesinger, Dianne P. O'Leary, and Jade Goldstein. *Back to Basics: CLASSY 2006*. U.S. National Inst. of Standards and Technology, 2006b.

Hoa T. Dang. Overview of DUC 2005. In *Document Understanding Conference*, 2005.

Hoa T. Dang. Overview of DUC 2007. In *Document Understanding Conference*, 2007.

Hoa Tran Dang. Overview of the tac 2008 opinion question answering and summarization tasks. In *Proc. TAC 2008*, 2008.

Van Dang. RankLib. Online, 2011.

Anirban Dasgupta, Ravi Kumar, and Sujith Ravi. Summarization through submodularity and dispersion. In *Proceedings of the 51st Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 1014–1022, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Hal Daumé, III and Daniel Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 305–312, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220214.

Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 519–528, New York, NY, USA, 2003. ACM. ISBN 1-58113-680-3. doi: 10.1145/775152.775226.

Sashka T. Davis, John M. Conroy, and Judith D. Schlesinger. Occams - an optimal combinatorial covering algorithm for multi-document summarization. In *ICDM Workshops*, pages 454–463, 2012.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure trees. In *LREC*, 2006.

Gianluca Demartini, Malik Muhammad Saad Missen, Roi Blanco, and Hugo Zaragoza. Entity summarization of news articles. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 795–796, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835620.

Xiaowen Ding, Bing Liu, and Philip S. Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and*

Data Mining, WSDM '08, pages 231–240, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-927-2. doi: 10.1145/1341531.1341561.

Bonnie J Dorr, David Zajic, and Richard Schwartz. Hedge trimmer: a parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop - Volume 5*, HLT-NAACL-DUC '03, pages 1 – 8, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics, Association for Computational Linguistics. doi: 10.3115/1119467.1119468.

Lan Du, Wray Buntine, and Huidong Jin. A segmented topic model based on the two-parameter poisson-dirichlet process. *Mach. Learn.*, 81:5–19, October 2010. ISSN 0885-6125. doi: http://dx.doi.org/10.1007/s10994-010-5197-4.

Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, July 1998. ISBN 0521629713.

Jacob Eisenstein, Noah A. Smith, and Eric P. Xing. Discovering sociolinguistic associations with structured sparsity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1365–1374, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Micha Elsner and Deepak Santhanam. Learning to fuse disparate sentences. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, pages 54–63, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 9781937284053.

Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as

salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, December 2004. ISSN 1076-9757.

Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06)*, pages 417–422, 2006.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4.

Richárd Farkas, Veronika Vincze, György Móra, János Csirik, and György Szarvas. The conll-2010 shared task: Learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning — Shared Task*, CoNLL '10: Shared Task, pages 1–12, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-84-8.

Song Feng, Jun Seok Kang, Polina Kuznetsova, and Yejin Choi. Connotation lexicon: A dash of sentiment beneath the surface meaning. In *ACL*, pages 1774–1784. The Association for Computer Linguistics, 2013. ISBN 978-1-937284-50-3.

Raquel Fernández, Matthew Frampton, John Dowding, Anish Adukuzhiyil, Patrick Ehlen, and Stanley Peters. Identifying relevant phrases to summarize decisions in spoken meetings. INTERSPEECH-2008, pages 78–81, 2008.

Raquel Fernández, Matthew Frampton, John Dowding, Anish Adukuzhiyil, Patrick Ehlen, and Stanley Peters. Identifying relevant phrases to summarize decisions in spoken meetings. In *INTERSPEECH*, pages 78–81, 2008.

Oliver Ferschke, Iryna Gurevych, and Yevgen Chebotar. Behind the article: Recognizing dialog acts in wikipedia talk pages. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 777–786, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. ISBN 978-1-937284-19-0.

Therese Firmin Hand. *Intelligent Scalable Text Summarization*, chapter A Proposal for Task-based Evaluation of Text Summarization Systems. 1997.

Radu Florian, Hany Hassan, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, Xiaoqiang Luo, Nicolas Nicolov, and Salim Roukos. A statistical model for multilingual entity detection and tracking. In *HLT-NAACL*, pages 1–8, 2004.

Matthew Frampton, Jia Huang, Trung Huu Bui, and Stanley Peters. Real-time decision detection in multi-party dialogue. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, pages 1133–1141, 2009. ISBN 978-1-932432-63-3.

Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res.*, 4:933–969, December 2003. ISSN 1532-4435.

Maria Fuentes, Enrique Alfonseca, and Horacio Rodríguez. Support vector machines for query-focused summarization trained and evaluated on pyramid data. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 57–60, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

Michel Galley. A skip-chain conditional random field for ranking meeting utterances by

importance. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 364–372, 2006. ISBN 1-932432-73-6.

Michel Galley and Kathleen McKeown. Lexicalized Markov grammars for sentence compression. NAACL '07, pages 180–187, Rochester, New York, April 2007. Association for Computational Linguistics.

Michel Galley, Kathleen McKeown, Julia Hirschberg, and Elizabeth Shriberg. Identifying agreement and disagreement in conversational speech: use of Bayesian networks to model pragmatic dependencies. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 669+, Morristown, NJ, USA, 2004. Association for Computational Linguistics. doi: 10.3115/1218955.1219040.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 340–348, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Wei Gao, Peng Li, and Kareem Darwish. Joint topic modeling for event summarization across news and social media streams. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1173–1182, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2398417.

Nikhil Garg, Benot Favre, Korbinian Riedhammer, and Dilek Hakkani-Tr. Clusterrank: a graph based method for meeting summarization. In *INTERSPEECH*, pages 1499–1502. ISCA, 2009.

Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. Abstractive summarization of product reviews using discourse structure. In *Proceed-*

ings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1602–1613, Doha, Qatar, October 2014. Association for Computational Linguistics.

G. Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, 2005.

Jennifer Gillenwater, Alex Kulesza, and Ben Taskar. Discovering diverse and salient threads in document collections. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 710–720, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Dan Gillick and Benoit Favre. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 10–18, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-35-0.

Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 121–128, New York, NY, USA, 1999. ACM. ISBN 1-58113-096-1. doi: 10.1145/312624.312665.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic Summarization - Volume 4*, NAACL-ANLP-AutoSum '00, pages 40–48, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. doi: 10.3115/1117575.1117580.

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*, SIGIR '01, pages 19–25, New York, NY, USA, 2001. ACM. ISBN 1-58113-331-6. doi: http://doi.acm.org/10.1145/383952.383955.

Joao Graca, Kuzman Ganchev, and Ben Taskar. Expectation maximization and posterior constraints. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 569–576. MIT Press, Cambridge, MA, 2008.

David Graff. English Gigaword Fifth Edition LDC2011T07. 2003.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.

Ben Hachey. Multi-document summarisation using generic relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 420–429, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-59-6.

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 '09, pages 362–370. Association for Computational Linguistics, 2009. ISBN 978-1-932432-41-1.

Sangyun Hahn, Richard Ladner, and Mari Ostendorf. Agreement/disagreement classification: Exploiting unlabeled data using contrast classifiers. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short*

Papers, pages 53–56, New York City, USA, June 2006. Association for Computational Linguistics.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278.

Ahmed Hassan, Vahed Qazvinian, and Dragomir Radev. What's with the attitude?: Identifying sentences with attitude in online discussions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1245–1255, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Ahmed Hassan, Amjad Abu-Jbara, and Dragomir Radev. Detecting subgroups in online discussions by modeling positive and negative relations among participants. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 59–70, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Michael Heilman and Noah A. Smith. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 609–617, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5.

Julia Hirschberg and Diane Litman. Empirical studies on the disambiguation of cue phrases. *Comput. Linguist.*, 19(3):501–530, September 1993. ISSN 0891-2017.

Eduard Hovy, Chin yew Lin, Liang Zhou, and Junichi Fukumoto. Automated summa-

rization evaluation with basic elements. In *In Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC*, 2006.

Pei-yun Hsueh and Johanna Moore. What decisions have you made: Automatic decision detection in conversational speech. In *In NAACL/HLT 2007*, 2007.

Meishan Hu, Aixin Sun, and Ee-Peng Lim. Comments-oriented document summarization: Understanding documents with readers' feedback. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 291–298, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390385.

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, pages 168–177, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. doi: 10.1145/1014052.1014073.

Jagadeesh Jagarlamudi, Prasad Pingali, and Vasudeva Varma. *Query Independent Sentence Scoring approach to DUC 2006*. 2006.

A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. The icsi meeting corpus. volume 1, pages I–364–I–367 vol.1, 2003.

Hongyan Jing, Regina Barzilay, Kathleen Mckeown, and Michael Elhadad. Summarization Evaluation Methods: Experiments and Analysis. In *In AAAI Symposium on Intelligent Summarization*, 1998.

Thorsten Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Machine*

Learning: ECML-98, volume 1398, chapter 19, pages 137–142. Berlin/Heidelberg, 1998. ISBN 3-540-64417-2. doi: 10.1007/BFb0026683.

Thorsten Joachims. Advances in kernel methods. chapter Making Large-scale Support Vector Machine Learning Practical, pages 169–184. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-19416-3.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pages 133–142, New York, NY, USA, 2002. ACM. ISBN 1-58113-567-X. doi: 10.1145/775047.775067.

Karen Sparck Jones. Automatic summarising: Factors and directions. In *Advances in Automatic Text Summarization*, pages 1–12. MIT Press, 1998.

Q. Jones and S. Rafaeli. Time to split, virtually: discourse architecture and community building create vibrant virtual publics. *Electronic Markets*, 10:214–223, 2000.

Mahesh Joshi and Carolyn Penstein-Rosé. Generalizing dependency features for opinion mining. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 313–316, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

George Karypis. CLUTO - a clustering toolkit. Technical Report #02-017, November 2003.

Hyun Duk Kim, Dae Hoon Park, V.G.Vinod Vydiswaran, and ChengXiang Zhai. Opinion summarization using entity features and probabilistic sentence coherence optimization: Uiuc at tac 2008 opinion summarization pilot. In *Proc. TAC 2008*, 2008.

J. Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and

Flesch Reading Ease Formula) for Navy Enlisted Personnel. Technical report, February 1975.

Aniket Kittur and Robert E. Kraut. Harnessing the wisdom of crowds in wikipedia: Quality through coordination. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work*, CSCW '08, pages 37–46, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-007-4. doi: 10.1145/1460563.1460572.

Aniket Kittur, Bongwon Suh, Bryan A. Pendleton, and Ed H. Chi. He says, she says: Conflict and coordination in wikipedia. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 453–462, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-593-9. doi: 10.1145/1240624.1240698.

Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1075096.1075150.

Jon Kleinberg and Eva Tardos. *Algorithm Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. ISBN 0321295358.

Kevin Knight and Daniel Marcu. Statistics-based summarization - step one: Sentence compression. AAAI '00, pages 703–710. AAAI Press, 2000. ISBN 0-262-51112-6.

Ioannis Konstas and Mirella Lapata. Concept-to-text generation via discriminative reranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 369–378, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

R. E. Kraut and P. Resnick. *Building successful online communities: Evidence-based social design*. MIT Press, Cambridge, MA, 2012.

Finley Lacatusu, Andrew Hickl, Kirk Roberts, Ying Shi, Jeremy Bensley, Bryan Rink, Patrick Wang, and Lara Taylor. *LCCs gistexter at duc 2006: Multi-strategy multi-document summarization*. 2006.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001a. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001b. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1.

J R Landis and G G Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

Mirella Lapata. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 545–552, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1075096.1075165.

Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. Sentiment summarization: Evaluating and learning user preferences. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 514–522, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 497–506, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557077.

P. M. Lewis, II and R. E. Stearns. Syntax-directed transduction. *J. ACM*, 15(3):465–488, July 1968. ISSN 0004-5411. doi: 10.1145/321466.321477.

Baoli Li, Yandong Liu, and Eugene Agichtein. Cocqa: Co-training over questions and answers with an application to predicting question subjectivity orientation. In *EMNLP*, pages 937–946, 2008a.

Chen Li, Fei Liu, Fuliang Weng, and Yang Liu. Document summarization via guided sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 490–500, Seattle, Washington, USA, October 2013a. Association for Computational Linguistics.

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)*, pages 1004–1013, Sofia, Bulgaria, August 2013b. Association for Computational Linguistics.

Chen Li, Yang Liu, Fei Liu, Lin Zhao, and Fuliang Weng. Improving multi-documents summarization by sentence compression based on expanded constituent parse trees. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 691–701, Doha, Qatar, October 2014. Association for Computational Linguistics.

Wenjie Li, You Ouyang, Yi Hu, and Furu Wei. Polyu at tac 2008. In *Proc. TAC 2008*, 2008b.

Chin-Yew Lin. Improving summarization performance by sentence compression: a pilot study. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages - Volume 11*, AsianIR '03, pages 1–8, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1118935.1118936.

Chin-Yew Lin and Eduard Hovy. The automated acquisition of topic signatures for text summarization. COLING '00, pages 495–501, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. ISBN 1-55860-717-X. doi: 10.3115/990820.990892.

Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 71–78, 2003. doi: http://dx.doi.org/10.3115/1073445.1073465.

Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 912–920, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 510–520, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-87-9.

S.-H. Lin, Y.-M. Yeh, and B. Chen. Leveraging kullback-leibler divergence measures and information-rich cues for speech summarization. 2010.

Shih-Hsiang Lin and Berlin Chen. A risk minimization framework for extractive speech summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 79–87, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012. doi: 10.2200/S00416ED1V01Y201204HLT016.

Fei Liu and Yang Liu. From extractive to abstractive meeting summaries: can it be done by sentence compression? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 261–264, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

Yuanjie Liu, Shasha Li, Yunbo Cao, Chin-Yew Lin, Dingyi Han, and Yong Yu. Understanding and summarizing answers in community-based question answering services. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 497–504, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. ISBN 978-1-905593-44-6.

Annie Louis and Ani Nenkova. Automatically assessing machine summary content without a gold standard. *Comput. Linguist.*, 39(2):267–300, June 2013. ISSN 0891-2017. doi: 10.1162/COLI_a_00123.

Bin Lu, Myle Ott, Claire Cardie, and Benjamin Tsou. Multi-aspect sentiment analysis with topic models. In *Workshop on Sentiment Elicitation from Natural Text for Information Retrieval and Extraction*, 2011.

H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2): 159–165, April 1958. ISSN 0018-8646. doi: 10.1147/rd.22.0159.

Xiaoqiang Luo and Imed Zitouni. Multi-lingual coreference resolution with syntactic features. In *HLT/EMNLP*, 2005.

Xiaoqiang Luo, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. A mention-synchronous coreference resolution algorithm based on the bell tree. In *ACL*, pages 135–142, 2004.

Xiaoqiang Luo, Hema Raghavan, Vittorio Castelli, Sameer Maskey, and Radu Florian. Finding what matters in questions. In *HLT-NAACL*, pages 878–887, 2013.

Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. Topic-driven reader comments summarization. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 265–274, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1156-4. doi: 10.1145/2396761.2396798.

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*, AAAI'97/IAAI'97, pages 622–628. AAAI Press, 1997. ISBN 0-262-51095-2.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland, 2014. Association for Computational Linguistics.

Yi Mao and Guy Lebanon. Isotonic conditional random fields and local sentiment flow. In *Advances in Neural Information Processing Systems*, 2007.

André F. T. Martins and Noah A. Smith. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Pro-*

gramming for Natural Langauge Processing, ILP '09, pages 1–9, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-35-0.

Sameer Maskey and Julia Hirschberg. Comparing Lexical, Acoustic/Prosodic, Structural and Discourse Features for Speech Summarization. In *Proc. European Conference on Speech Communication and Technology (Eurospeech)*, 2005.

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

Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

I. Mccowan, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. The ami meeting corpus. In *In: Proceedings Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research. L.P.J.J. Noldus, F. Grieco, L.W.S. Loijens and P.H. Zimmerman (Eds.), Wageningen: Noldus Information Technology*, 2005.

Ryan McDonald. Discriminative Sentence Compression with Soft Syntactic Constraints. In *Proceedings of the 11th˜EACL*, Trento, Italy, April 2006.

Ryan McDonald. A study of global inference algorithms in multi-document summarization. ECIR'07, pages 557–564, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-71494-1.

Kathleen McKeown, Rebecca J. Passonneau, David K. Elson, Ani Nenkova, and Julia Hirschberg. Do summaries help? In *Proceedings of the 28th Annual International*

ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05, pages 210–217, New York, NY, USA, 2005. ACM. ISBN 1-59593-034-5. doi: 10.1145/1076034.1076072.

Yashar Mehdad, Giuseppe Carenini, and Raymond T. Ng. Abstractive summarization of spoken and written conversations based on phrasal queries. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1220–1230, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pages 775–780. AAAI Press, 2006. ISBN 978-1-57735-281-5.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38:39–41, November 1995. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/219717.219748.

Gilad Mishne and Natalie Glance. Leave a reply: An analysis of weblog comments. In *Third annual workshop on the Weblogging ecosystem*, 2006.

Amita Misra and Marilyn Walker. Topic independent identification of agreement and disagreement in social media dialogue. In *Proceedings of the SIGDIAL 2013 Conference*, pages 41–50, Metz, France, August 2013. Association for Computational Linguistics.

Michael Mozer, Michael I. Jordan, and Thomas Petsche, editors. *Advances in Neural*

Information Processing Systems 9, NIPS, Denver, CO, USA, December 2-5, 1996, 1997. MIT Press.

Gabriel Murray. Abstractive meeting summarization as a markov decision process. In Denilson Barbosa and Evangelos Milios, editors, *Advances in Artificial Intelligence*, volume 9091 of *Lecture Notes in Computer Science*, pages 212–219. Springer International Publishing, 2015. ISBN 978-3-319-18355-8. doi: 10.1007/978-3-319-18356-5_19.

Gabriel Murray and Giuseppe Carenini. Summarizing spoken and written conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 773–782, 2008.

Gabriel Murray, Steve Renals, and Jean Carletta. Extractive summarization of meeting recordings. In *in Proceedings of the 9th European Conference on Speech Communication and Technology*, pages 593–596, 2005a.

Gabriel Murray, Steve Renals, and Jean Carletta. Extractive summarization of meeting recordings. In *INTERSPEECH*, pages 593–596, 2005b.

Gabriel Murray, Giuseppe Carenini, and Raymond Ng. Interpretation and transformation for abstracting conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 894–902, Stroudsburg, PA, USA, 2010a. Association for Computational Linguistics. ISBN 1-932432-65-5.

Gabriel Murray, Giuseppe Carenini, and Raymond T. Ng. Generating and validating abstracts of meeting conversations: a user study. In *INLG'10*, 2010b.

Ramesh Nallapati, Ao Feng, Fuchun Peng, and James Allan. Event threading within news topics. In *Proceedings of the Thirteenth ACM International Conference on*

Information and Knowledge Management, CIKM '04, pages 446–453, New York, NY, USA, 2004. ACM. ISBN 1-58113-874-1. doi: 10.1145/1031171.1031258.

S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, March 1970. ISSN 0022-2836.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functionsI. *Mathematical Programming*, 14(1):265–294, December 1978. ISSN 0025-5610. doi: 10.1007/bf01588971.

Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The pyramid method. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 145–152, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

Ani Nenkova and Lucy Vanderwende. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*, 2005.

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*, SIGIR '06, pages 573–580, New York, NY, USA, 2006. ACM. ISBN 1-59593-369-7. doi: 10.1145/1148170.1148269.

Jun-Ping Ng, Yan Chen, Min-Yen Kan, and Zhoujun Li. Exploiting timelines to enhance multi-document summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 923–933, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2010.

Jahna Otterbacher, Güneş Erkan, and Dragomir R. Radev. Using random walks for question-focused sentence retrieval. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 915–922, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1220575.1220690.

You Ouyang, Wenjie Li, Sujian Li, and Qin Lu. Applying regression models to query-focused multi-document summarization. *Inf. Process. Manage.*, 47(2):227–237, March 2011. ISSN 0306-4573. doi: 10.1016/j.ipm.2010.03.005.

P. Over and J. Yen. An introduction to DUC 2003: Intrinsic evaluation of generic news text summarization systems, 2003.

Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008. ISSN 1554-0669. doi: 10.1561/1500000011.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118693.1118704.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–

318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135.

Michael J. Paul, ChengXiang Zhai, and Roxana Girju. Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 66–76, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Daniele Pighin, Marco Cornolti, Enrique Alfonseca, and Katja Filippova. Modelling events through memory-based, open-ie patterns for abstractive summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 892–901, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

Prasad Pingali, Rahul K, and Vasudeva Varma. *IIIT Hyderabad at DUC 2007*. U.S. National Inst. of Standards and Technology, 2007.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. The penn discourse treebank 2.0. In *LREC*. European Language Resources Association, 2008.

Minghui Qiu, Liu Yang, and Jing Jiang. Mining user relations from online discussions using sentiment analysis and probabilistic matrix factorization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 401–410, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

Dragomir R. Radev and Kathleen R. McKeown. Generating natural language summaries from multiple on-line sources. *Comput. Linguist.*, 24(3):470–500, September 1998. ISSN 0891-2017.

Peter A. Rankel, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. A decade of automatic content evaluation of news summaries: Reassessing the state of the art. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 131–136, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. *Building natural language generation systems*. Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-62036-8.

Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tür. Long story short - global unsupervised models for keyphrase based meeting summarization. *Speech Commun.*, 52(10):801–815, October 2010. ISSN 0167-6393. doi: 10.1016/j.specom.2010.06.002.

Alan Ritter, Mausam, Oren Etzioni, and Sam Clark. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1104–1112, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339704.

Daniel M. Romero, Wojciech Galuba, Sitaram Asur, and Bernardo A. Huberman. Influence and passivity in social media. In *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, pages 113–114, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0637-9. doi: 10.1145/1963192.1963250.

W. Sack. Digital formations: It and new architectures in the global realm. chapter Discourse architecture and very large-scale conversation, pages 242–282. Princeton University Press, Princeton, NJ USA, 2005.

Gerard Salton and Michael J McGill. Introduction to modern information retrieval. 1986.

Oana Sandu, Giuseppe Carenini, Gabriel Murray, and Raymond Ng. Domain adaptation to summarize human conversations. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, DANLP 2010, pages 16–22, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-80-0.

Venu Satuluri, Srinivasan Parthasarathy, and Yiye Ruan. Local graph sparsification for scalable clustering. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 721–732, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0661-4. doi: 10.1145/1989323.1989399.

E. A. Schegloff and H. Sacks. Opening up closings. *Semiotica*, 8(4):289–327, 1973.

K.A. Schriver. Evaluating text quality: the continuum from text-focused to reader-focused methods. *Professional Communication, IEEE Transactions on*, 32(4):238–255, Dec 1989.

Dafna Shahaf and Carlos Guestrin. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 623–632, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1. doi: 10.1145/1835804.1835884.

Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. Trains of thought: Generating information maps. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 899–908, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187957.

Chao Shen and Tao Li. Learning to rank for query-focused multi-document summarization. In Diane J. Cook, Jian Pei, Wei Wang 0010, Osmar R. Zaane, and Xindong Wu, editors, *ICDM*, pages 626–634. IEEE, 2011. ISBN 978-0-7695-4408-3.

Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. Large-margin learning of submodular summarization models. EACL '12, pages 224–233, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. ISBN 978-1-937284-19-0.

Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004. ISSN 0960-3174. doi: 10.1023/B:STCO.0000035301.49549.88.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning Syntactic Patterns for Automatic Hypernym Discovery. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press, Cambridge, MA, 2005.

Swapna Somasundaran and Janyce Wiebe. Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 226–234, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-45-9.

Keith E Stanovich. Matthew effects in reading: Some consequences of individual differences in the acquisition of literacy. *Reading research quarterly*, pages 360–407, 1986.

Andreas Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 2, pages 901–904, Denver, USA, 2002.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge, MA, 1966.

Veselin Stoyanov and Claire Cardie. Partially supervised coreference resolution for opinion summarization through structured rule learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 336–344, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. ISBN 1-932432-73-6.

Joel R Tetreault and Martin Chodorow. The ups and downs of preposition error detection in esl writing. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 865–872. Association for Computational Linguistics, 2008.

Kapil Thadani and Kathleen McKeown. Supervised sentence fusion with single-stage inference. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 1410–1418, 2013.

Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 327–335, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. ISBN 1-932432-73-6.

Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 111–120. ACM, 2008. ISBN 978-1-60558-085-2. doi: http://doi.acm.org/10.1145/1367497.1367513.

Mattia Tomasoni and Minlie Huang. Metadata-aware measures for answer summarization in community question answering. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 760–769, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Kristina Toutanova, Chris Brockett, Michael Gamon, Jagadeesh Jagarlamudi, Hisami Suzuki, and Lucy Vanderwende. The PYTHY Summarization System: Microsoft Research at DUC 2007. In *Proc. of DUC*, 2007.

Jenine Turner and Eugene Charniak. Supervised and unsupervised learning for sentence compression. ACL '05, pages 290–297, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219876.

Andrew Turpin, Yohannes Tsegay, David Hawking, and Hugh E. Williams. Fast generation of result snippets in web search. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, pages 127–134, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277766.

Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 777–785, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5.

Ba-Quy Vuong, Ee-Peng Lim, Aixin Sun, Minh-Tam Le, Hady Wirawan Lauw, and Kuiyu Chang. On ranking controversies in wikipedia: Models and evaluation. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 171–182, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-927-2. doi: 10.1145/1341531.1341556.

Marilyn Walker, Jean Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. A corpus for research on deliberation and debate. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul,

Turkey, may 2012a. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Marilyn A. Walker, Owen Rambow, and Monica Rogati. Spot: a trainable sentence planner. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, pages 1–8, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics. doi: 10.3115/1073336.1073339.

Marilyn A. Walker, Pranav Anand, Rob Abbott, and Ricky Grant. Stance classification using dialogic properties of persuasion. In *HLT-NAACL*, pages 592–596. The Association for Computational Linguistics, 2012b. ISBN 978-1-937284-20-6.

Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In *In SIGIR 08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314. ACM, 2008.

Lu Wang and Claire Cardie. Summarizing decisions in spoken meetings. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, pages 16–24, Portland, Oregon, June 2011. Association for Computational Linguistics.

Lu Wang, Hema Raghavan, Claire Cardie, and Vittorio Castelli. Query-Focused Opinion Summarization for User-Generated Content. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1660–1669, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

Lu Wang, Claire Cardie, and Galen Marchetti. Socially Informed Timeline Generation

for Complex Events. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2015.

Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.

Wen Wang, Sibel Yaman, Kristin Precoda, Colleen Richey, and Geoffrey Raymond. Detection of agreement and disagreement in broadcast conversations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 374–378, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-88-6.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1220575.1220619.

Pengtao Xie and Eric Xing. Integrating document clustering and topic modeling. In *Proceedings of the Twenty-Ninth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-13)*, pages 694–703, Corvallis, Oregon, 2013. AUAI Press.

Shasha Xie, Yang Liu, and Hui Lin. Evaluating the effectiveness of features and sampling in extractive meeting summarization. In *in Proc. of IEEE Spoken Language Technology (SLT*, 2008.

Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. Evolutionary timeline summarization: A balanced optimization framework via it-

erative substitution. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 745–754, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2010016.

Zi Yang, Keke Cai, Jie Tang, Li Zhang, Zhong Su, and Juanzi Li. Social context summarization. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 255–264, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2009954.

Taha Yasseri, Róbert Sumi, András Rung, András Kornai, and János Kertész. Dynamics of conflicts in wikipedia. *CoRR*, abs/1202.3643, 2012.

Jie Yin, Paul Thomas, Nalin Narang, and Cecile Paris. Unifying local and global agreement and disagreement classification in online debates. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '12, pages 61–69, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2003.

David Zajic, Bonnie J Dorr, Jimmy Lin, and R. Schwartz. Sentence compression as a component of a multi-document summarization system. *Proceedings of the 2006 Document Understanding Workshop, New York*, 2006.

Klaus Zechner. Automatic summarization of open-domain multiparty dialogues in diverse genres. *Comput. Linguist.*, 28:447–485, December 2002.

Tong Zhang and Frank J. Oles. Text categorization based on regularized linear classification methods. *Inf. Retr.*, 4(1):5–31, April 2001. ISSN 1386-4564. doi: 10.1023/A:1011441423217.

Xin Wayne Zhao, Yanwei Guo, Rui Yan, Yulan He, and Xiaoming Li. Timeline generation with social attention. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 1061–1064, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2034-4. doi: 10.1145/2484028.2484103.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. In *Technical Report CMU-CALD-02-107*, 2002.