IDENTIFYING THE GIST OF CONVERSATIONAL TEXT:

AUTOMATIC KEYWORD EXTRACTION AND SUMMARIZATION

by

Fei Liu

APPROVED BY SUPERVISORY COMMITTEE:

Dr. Yang Liu, Chair

Dr. Carlos Busso

Dr. Sanda Harabagiu

Dr. Vincent Ng

Dedicated to my family.

IDENTIFYING THE GIST OF CONVERSATIONAL TEXT:

AUTOMATIC KEYWORD EXTRACTION AND SUMMARIZATION

by

FEI LIU, B.S., M.S.

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

May, 2011

PREFACE

This dissertation was produced in accordance with guidelines which permit the inclusion as part of the dissertation the text of an original paper or papers submitted for publication. The dissertation must still conform to all other requirements explained in the "Guide for the Preparation of Master's Theses and Doctoral Dissertations at The University of Texas at Dallas." It must include a comprehensive abstract, a full introduction and literature review, and a final overall conclusion. Additional material (procedural and design data as well as descriptions of equipment) must be provided in sufficient detail to allow a clear and precise judgment to be made of the importance and originality of the research reported.

It is acceptable for this dissertation to include as chapters authentic copies of papers already published, provided these meet type size, margin, and legibility requirements. In such cases, connecting texts which provide logical bridges between different manuscripts are mandatory. Where the student is not the sole author of a manuscript, the student is required to make an explicit statement in the introductory material to that manuscript describing the student's contribution to the work and acknowledging the contribution of the other author(s). The signature of the Supervising Committee which precedes all other material in the dissertation attest to the accuracy of this statement.

ACKNOWLEDGMENTS

I would like to thank all my friends, colleagues, professors, and staffs in the Human Language Technology Research Institute and Erik Jonsson School of Engineering & Computer Science at the University of Texas at Dallas.

Special thanks to my advisor Prof. Yang Liu, who offered great guidance on my research and study. Yang, I would not have been able to accomplish this without your help and support throughout the years. You have great knowledge on speech and language processing, yet are so patient and always willing to help us on any research problem, big or small; discussions with you have sparked many new thoughts and ideas. I also appreciate your efforts in promoting and attracting female students to the field. I'm very fortunate to have you as my advisor, and more importantly, as a role model that gives me guidance and inspiration throughout the life.

I would like to thank Prof. Lide Wu and Prof. Xuanjing Huang for their supervision in my undergraduate and master studies in Fudan University, Shanghai, China. Thank you for guiding me to the fantastic world of natural language processing and machine learning.

Many thanks also go to my dissertation committee members, Prof. Sanda Harabagiu, Prof. Vincent Ng, and Prof. Carlos Busso. I'm very grateful to have you as my dissertation committees. Thank you all for the nice suggestions and feedbacks on my dissertation, and for the great advice that will benefit my life and future career.

I'm very fortunate to have the opportunity to work together with many great colleagues in my PhD study, you are the best lab folks I can imagine. Thank Mark Hittinger, our lab admin, for answering all my Linux configuration questions. Thank Feifan Liu and Thamar Solorio, who were working in our lab as postdoctoral fellows, for the helpful discussions, advice, and collaborations. My appreciation also goes to other colleagues and fellow students: Shasha Xie, Deana Pennell, Melissa Sherman, Bin Li, Je Hun Jeon, Keyur Gabani, Zhonghua Qu, Dong Wang, Khairun-nisa Hassanali,

Rui Xia, Justin Schneider, and Duc Le. Thank you all for giving me so many nice memories and for making our lab an attractive place to work. My thanks also go to the ex-colleagues in the Media Computing and Web Intelligence Lab in Fudan University for their generous help and support.

During the summer and fall of 2010, I worked as a research intern in Bosch Reserach & Technology Center, Palo Alto, CA. My special thanks to the senior manager Fuliang Weng for offering me all the guidance in work and study. Fuliang, your insightful words, advice, and guidance have greatly enlightened my life and will keep benefiting my future career. I would also thank the regional president Horst Muenzel, director Hauke Schmidt, HR manager Shauna Zimmerman, and many great colleagues in the UI team (in alphabetical order): Jens Faenger, Zhe Feng, Madhuri Raya, Liu Ren, Zhongnan Shen, Soundar Srinivasan, Baoshi Yan, interns Haidong Chen, Yueqi Hu, Yize Li, Bingqing Wang, and many other colleagues in Bosch RTC.

Last, I would like to thank my parents and my husband for their relentless support through the ups and downs of my life. I thank my family for their optimistic attitude towards life, for the many good habits they taught me from a very early age, and for their endless love and support.

March, 2011

IDENTIFYING THE GIST OF CONVERSATIONAL TEXT:
AUTOMATIC KEYWORD EXTRACTION AND SUMMARIZATION

Publication No. _____

Fei Liu, Ph.D.
The University of Texas at Dallas, 2011

Supervising Professor: Dr. Yang Liu

With the rapid development of communication technologies and mass storage techniques, many conversational texts have quickly emerged as significant information sources, such as emails, forums, meeting conversation transcripts, chat logs, microblogs, etc. The ability to identify the gist of these conversational texts enables us to quickly browse through the huge amount of data and obtain the essential information. On the other hand, the conversational text style also poses great challenges to the traditional language processing techniques, including redundancies, disfluencies, ill-formed sentence structure, high word error rates, and so forth. In this work, we focus on keyword extraction and summarization on meeting transcripts, and also explore summarizing the Twitter posts (tweets) as another domain of conversational text.

We propose to extract keywords using a novel supervised framework that incorporates various knowledge sources: beyond the traditional widely used features (e.g., TF-IDF, position information), we introduce additional rich features including term specificity information, decision-making

sentence related features, speaker and prominence based features, and features extracted from system generated summaries. We propose a feedback strategy to reinforce the impact of summary sentences on selecting effective keywords. We conduct analysis to evaluate feature effectiveness using different feature selection processes, and define various measurements to characterize the quality of summaries that can benefit the keyword extraction task. We also evaluate system performance using both human transcripts and different automatic speech recognizer (ASR) output (1-best and n-best), and show promising improved keyword extraction results using n-best ASR output over 1-best hypothesis.

For extractive meeting summarization, we explore multiple meeting-specific characteristics. We propose to use topic labels and speaker-dependent characteristics (such as verboseness, gender, native language, role in the meeting) to improve extractive meeting summarization performance. These properties were incorporated in both unsupervised Maximum Marginal Relevance (MMR) approach and the supervised framework. We observe consistent improvements using our proposed approaches, on both human transcripts and ASR output, and using different evaluation metrics including ROUGE, Pyramid, and a DA-level F-measure score.

Beyond extractive summarization, we propose to perform sentence compression on the extractive summary to improve its readability and make it more like an abstractive summary. Various automatic compression algorithms are investigated, including the integer linear programming (ILP) based approach with filler phrase detection, a noisy-channel approach using Markovization formulation of grammar rules, as well as the conditional random fields (CRF) based approach. The automatically compressed utterances are compared against both human compression and the abstractive summaries. We also evaluate the impact of using compressed utterances on summariza-

tion, and propose a fully automatic summarizer that generates compressed meeting summaries by combing the utterance compression module with an extractive summarization system.

We perform exploratory summarization studies on another domain of conversational text – the Twitter posts, to help users quickly browse through any available topics. As an important first step, we propose a novel letter transformation approach to convert the nonstandard tokens in the tweets into standard English words. Different from the prior work, our approach requires neither pre-categorization nor human supervision. The approach models the generation process from the dictionary words to nonstandard tokens under a sequence labeling framework. We also explore summarizing the Twitter topics using the concept-based global optimization approach, and investigate the effect of both noisy nonstandard tokens and linked web contents on the summarization performance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Automatic keyword extraction aims to select a set of representative words that can signature the main topics of the source text; while summarization can convey more information by providing the users with extracted salient sentences or produce an abstractive description using language generation techniques. Both keyword extraction and summarization can help users quickly browse through a large amount of documents and have been actively studied in the past decades within the natural language processing community. Related research mainly focuses on the formal written text, including extracting keywords/keyphrases from journal papers, news documents, scientific articles, biomedical text, etc., as well as summarizing newswires and blogs.

Yet with the fast developed information technology and mass storage techniques, the users' need for quickly navigating and accessing information has gone far beyond the traditional news articles. Nonconventional information communication technologies, such as emails, forums, Twitter messages, recorded meeting conversations, have quickly emerged as significant information sources. The text created from these information sources bears the same conversational nature, and is substantially different from the written documents. For example, the traditional news articles are typically written by a single professional writer; the contents are well-structured with title, subtitle, paragraph or section information; the first paragraph usually signifies the author's

main point, and each following section introduces a new piece of relevant information or provides supporting arguments; the sentences in news articles are long and well-formed; word usage is standard with very rare spelling errors. In contrast, the conversational text exhibits very different characteristics: the contents are typically contributed by multiple participants, contain lots of redundancies and disfluencies, and have low text coherence; the sentences can be incomplete and ill-formed with colloquial expressions and grammatical errors; there is little structure information such as title or sentence boundary; word usage is very informal, with lots of slangs, acronyms, abbreviations, emoticons, etc.

Although processing the informal conversational text poses great challenges to the existing technologies, it is highly rewarding to extract gist information from these resources. Take the meeting domain for an example, browsing through previous business meetings can help a newcomer quickly become familiar with the work environment and the company culture; meeting participants might need to review the decisions made in the last meeting and progress on that basis; they might also want to know whether an issue has been discussed and solved in previous meetings; reviewing the meetings between doctors and patients can help doctors understand problems of a patient and improve the doctor-patient communication. In all these situations, the automatically extracted keywords and summaries can help users quickly browse through the lengthy meeting recordings and grab the gist in a short time. As another source of massive amount of conversational text, the social networking websites such as Twitter.com have quickly emerged in very recent years. According to the news release in September 2010 [1], Twitter has more than 145 million users in the world, and is generating on average 90 millions tweets per day. As a result, Twitter has become a significant source of gathering real-time information on almost any topic imaginable. Yet there

still lacks a systematical approach to automatically extract important information from the noisy conversational text style or summarize a specific topic in real-time.

In this work, we focus on keyword extraction and summarization using the meeting corpus, and investigate summarizing the Twitter topics as an exploratory study. For keyword extraction, our goal is to automatically select a set of representative words from the meeting transcripts that can effectively capture the main contents; for extractive meeting summarization, the goal is to extract a collection of salient sentences from the meeting recordings and concatenate them to form a coherent summary. On top of the extractive summaries, we further investigate if we could perform sentence compression on the extracted utterances, and make the resulting summaries more like abstracts. We also perform preliminary studies on summarizing the Twitter topics. We explore a variety of input text sources to examine the effect of noisy nonstandard tokens and linked web contents on the summarization performance.

1.2 Challenges Using Conversational Text

Meeting recordings bear many characteristics that are different from the traditional written text. In the following, we show an example of the meeting dialogue segment, which corresponds to 1.5 minutes of meeting speech. The human transcripts with hand-labeled sentence boundaries are shown in Table 1.1. Table 1.2 shows the output from an automatic speech recognizer (ASR) with pause based segmentation. We can see that meeting transcripts differ from written text significantly. Many successful language processing techniques have reported performance degradation on meeting transcripts. The following lists a few differences that may have a negative impact on the traditional keyword extraction and summarization systems:

Table 1.1. Sample meeting dialogue segment, transcribed by human subjects and annotated with sentence boundaries. Human annotated keywords are shown in bold.

ID	Speaker	Dialogue Act
423	me010	there there are a variety of ways of doing it
424	me010	uh let me just mention something that i don't want to pursue today
425	me010	which is there are technical ways of doing it
426	me010	uh i- i slipped a paper to bhaskara and about noisy-or's and noisy-maxes
427	me010	and
428	me010	there're ways to uh sort of back off on the purity of your bayes-net-edness
429	me003	mmm
430	me010	uh so if you co- you could ima- and i-
431	me010	now i don't know that any of those actually apply in this case
432	me010	but there is some technology you could try to apply
433	me003	so it's possible that we could do something like a summary node of some sort that
434	me010	yeah
435	me003	ok
436	me010	yeah
437	me010	and um
438	me010	so
439	me003	so in that case the sum- we'd have we
440	me003	i mean these wouldn't be the summary nodes
441	me003	we'd have the summary nodes like
442	me003	where the things were i guess maybe if thi- if things were related to business or some other
443	me010	yeah
444	me010	so what i was gonna say is is maybe a good at this point is to try to informally
445	me003	yeah
446	me010	i mean not necessarily in th- in this meeting but to try to informally think about what the **decision variables** are
447	me010	so if you have some bottom line uh **decision** about which mode
448	me010	you know what are the most relevant things
449	me003	mmm
450	me010	and the other trick which is not a technical trick it's kind of a knowledge engineering trick is to make the n- -pau- each node sufficiently narrow that you don't get this **combinatorics**
451	me010	so that if you decided that you could characterize the **decision** as a trade-off between three factors whatever they may be

Table 1.2. Sample meeting dialogue segment, transcribed by automatic speech recognizer (1-best output) with pause based segmentation.

ID	Utterance Segment
64	they're they're variety of ways of doing it uh let me just mention something that i don't want to pursue today which is there are technical ways of doing it uh i so i slipped the paper to bust corrupt and about noisy oars and always the taxes and their ways to uh sort of back off on the purity of your base net in this uh self if you could do could have matt and no i don't know that any of those actually apply in this case but there is some technology you could try to apply
65	uhhuh
66	so it's possible that we could do something like a summary note of some sort that okay
67	yeah
68	yeah
69	and um
70	so
71	so in that case the summer we'd have we'd i mean these wouldn't be the summer has we've had some and i was like weather things for i guess maybe taking things were related to business or
72	that so what i was going to say is is maybe a good idea at this point is to try to in the formally i mean not necessarily in the in this meeting that to try to informally think about what's the decision variables are so if you had some bottom line uh decision about which mode you know what are the most relevant things
73	some other yeah
74	uhhuh
75	and the other tricks which is not a technical ticket scuds and all that engineering trick is too makes the each note sufficiently narrow that you don't get this common at work so if you decided that you could characterize the decision is a trade off between three factors whatever they may be

1. Lexical density, measured using the percentage of content words, is low for meeting transcripts. According to [2], two content words per clause are quite typical in unplanned spoken text; in contrast, written text can often have around four to six content words (or even more) per clause. Fewer content words pose a main problem to many traditional language processing techniques.

2. Meeting transcripts lack structure information, such as title, paragraph, topic or sentence boundaries. Automatic sentence boundary detection and topic segmentation are still under development in the meeting domain. The pause based segmentation as shown Table 1.2 can only result in large blocks of text.

3. Sentences in meetings are often poorly structured. There are many incomplete sentences, interruptions, and disfluencies.

4. Multiple participants in meetings introduce new challenges. For example, different people have different speaking styles and word usage, participants also have different roles in topic discussions, and each participant can begin a new topic when starting his/her turn. These phenomena do not exist in most text domains where a document is generally written by one person (note there are text domains that have multiple authors, such as forum data). Therefore for multiparty meetings, the existing language processing technologies may need to incorporate speaker dependent features to account for different speakers.

5. High ASR error rate posts another challenge for processing meeting transcripts. Misrecognizing or missing important content words changes the desired word frequency. In addition, word errors significantly degrade many syntactic and semantic analysis techniques, such as

part-of-speech (POS) tagging and parsing, thus making it more difficult for deeper understanding of the data.

6. In meetings, the information flow among different participants can be accomplished via the combination of several modalities: depending on the literal utterances of the speakers; depending on the tone and prosody they used; and also depending on the visual evidence or so-called "social signals": such as head movement, eye focus, body gesture, etc. In the current work, we focus on extracting information from the spoken audio documents and their transcripts, therefore might lose important cues from other modalities.

Compared to meeting transcripts, the Twitter posts[1] are a largely unexploited genre. As a newly emerged service, Twitter provides a general platform for people to broadcast personalized news, share their opinions with others, monitor a situation or topic, etc. The Twitter posts exhibit many similar conversational text properties due to the inherent conversational nature and the length constraint. Table 1.3 shows example tweets related to the topic "Chilean miners". Several of the observed characteristics are listed in below:

1. All tweets are limited to 140 characters. Some tweets are news headlines from the official media, others are generated by users with various degrees of familiarity with the social media. The resulting tweets can be very different regarding the text quality and word usage.

2. Similar to the meeting transcripts, Twitter posts lack structure information, contain various ill-formed sentences and grammatical errors, and are contributed by various users. Regarding

[1]Throughout this work, we will use Twitter posts, Twitter messages, and tweets interchangeably to denote the status updates from the social networking website Twitter.com.

Table 1.3. Example Twitter posts collected for the topic “Chilean miners”

Twitter Topic: Chilean miners	
(1)	I <3 that those miners got rescued that is sooo amazing! Such an uplifting story better than what we hear most the time :) yay chilean miner
(2)	RT @WeHateDemi: Sooo happy for the Chilean miners!!!!!
(3)	RT @BreakingNews: More on Chilean mine breakthrough: Rescuers now determining safest way to pull miners out http://bit.ly/9zItP5
(4)	Rescuers Break Through To Trapped Chilean Miners http://bit.ly/blVn9p #chileanminers
(5)	@Pr1mr0se Excellent - I’ve only just come back onto Twitter after this morning - lots of Chilean Miners stuff! :-)
(6)	yaay chilean miners are rescueeed
(7)	dang feelin sori for th wives nd chikitaz of th 33 chilean miners. alot of fantasies made reality wen thy rturn #akulalwa
(8)	RT @Greytdog: BREAKTHROUGH!!! BREAKTHROUGH!!! Drill breaks through to trapped Chilean Miners http://bit.ly/choXVB

the word usage, there are lots of noisy nonstandard tokens, such as abbreviations (“feelin” for “feeling”), substitutions (“Pr1mr0se” for “Primrose”), slangs (“dang” for “damn”), foreign tokens (“akulalwa”), and emoticons (“<3” for love, “:-)” for smiley face).

3. Since the pure text can not carry prosody information, the Twitter users adopt many creative ways to emphasize things or express their emotions, including repeating letters (“sooo”, “yaay”, “rescueeed”), using multiple consecutive punctuation marks (“!!!!!”), capitalizing the first letter or all letters of some words, or using emoticons (“:-)”). These expressions are expected to work in a similar way as the speech prosody.

4. Twitter invented its own markup language, including the reply symbol “@” and the hashtag “#”. “@user” is used to reply to a specific user or call for attentions. “RT @user” means to cite a post exactly from the user. The hashtag “#topic” aims to label the post with topic

information, to help cluster together posts with similar topics. Both the reply symbol ("@") and hashtag ("#") can play a syntactic role in the sentences.

5. Tweets frequently contain embedded URLs that direct users to other online content, such as news web pages, blogs, organization homepages, etc. [3]. According to Twitter's news release on September 2010 [1], 25% of tweets contain an URL. These linked web pages provide a much richer source of information than what is possible in the 140-character tweet.

1.3 Contribution of the Proposed Work

In this work, a number of novel approaches are proposed for identifying the gist of conversational text. To facilitate the meeting browsing process, we propose to extract keywords using a novel supervised framework, utilizing meeting-specific characteristics for extractive summarization, as well as performing sentence compression on extractive summaries to generate more condensed summary representations. We explore different text sources for summarizing the Twitter topics, and investigate the effect of noisy contents on summarization performance by using Twitter posts with and without text normalization. The contributions of this dissertation are:

- For keyword extraction, we focus on the supervised framework and incorporate various knowledge sources: beyond the traditionally widely used features (e.g., TF-IDF, position information), we introduce additional rich features including term specificity information, decision-making sentence related features, speaker and prominence based features, and features extracted from system generated summaries. We propose a feedback strategy to reinforce the impact of summary sentences on selecting effective keywords. We conduct analysis

to evaluate feature effectiveness using different feature selection processes, and define various measurements to characterize the quality of summaries that can benefit the keyword extraction task. We also evaluate system performance using both human transcripts and different ASR output (1-best and n-best), and show promising improved keyword extraction results using n-best ASR output over 1-best hypothesis.

- For extractive meeting summarization, we explore multiple meeting-specific characteristics. We propose to use topic labels and speaker-dependent characteristics (such as verboseness, gender, native language, role in the meeting) to improve extractive meeting summarization performance. These properties were incorporated in both unsupervised Maximum Marginal Relevance (MMR) approach and the supervised framework. We observe consistent improvements using our proposed approaches, on both human transcripts and ASR output, and using different evaluation metrics including ROUGE, Pyramid, and a DA-level F-measure score.

- Beyond extractive summarization, we propose to perform sentence compression on the extractive summary to improve its readability and make it more like an abstractive summary. Compressing sentences could be a first step toward our ultimate goal of creating an abstract for spoken documents. We investigate various automatic compression algorithms, including the integer linear programming (ILP) based approach with filler phrase detection, a noisy-channel approach using Markovization formulation of grammar rules, as well as the conditional random fields (CRF) based approach. We perform large scale utterance compression annotation using the Amazon Mechanical Turk, and compare the automatically compressed utterances against human compression as well as the abstractive summaries. We also evaluate the impact of using compressed utterances on summarization, and propose a fully auto-

matic summarizer that generates compressed meeting summaries by combing the utterance compression module with an extractive summarization system. Our experiments show that compressing extractive summaries can improve human readability and the ROUGE scores against the original uncompressed extractive summaries.

- We perform exploratory Twitter topic summarization, to help users quickly browse through any available topics. As an important first step, we propose a novel letter transformation approach to convert the nonstandard tokens in the Twitter posts into standard English words. Our approach requires neither pre-categorization nor human supervision. It models the generation process from the dictionary words to nonstandard tokens under a sequence labeling framework, where each letter in the dictionary word can be retained, removed, or replaced by other letters/digits. To avoid the expensive and time consuming hand labeling process, we automatically collected a large set of noisy training pairs using a novel web-based approach, and aligned them at the character level for modeling training. For Twitter topic summarization, we focus on two questions that are not studied in previous literature: (1) Is the web content linked from the tweets useful for summarization? Can we integrate different text sources, including the tweets and linked web pages, to generate more informative Twitter topic summaries? (2) what is the effect of nonstandard tokens on the summarization performance? Will the summaries be improved if the noisy tweets were pre-normalized into standard English sentences? We investigate these two questions under a concept-based summarization framework using integer linear programming (ILP). We utilize text input that has various quality and is originated from multiple sources, and thoroughly analyze the resulting summaries using both automatic and human evaluation metrics.

CHAPTER 2

LITERATURE REVIEW

This chapter briefly summarizes previous work on keyword extraction, extractive and abstractive summarization. Most previous studies focus on the written text domain; however, there has been an increasing number of studies that tailor the tasks to the speech genre. In the most recent years, microblog summarization has also emerged to summarize the vast amount of user-generated contents on the web.

2.1 Keyword Extraction

Most of the related work for keyword extraction has been performed on the written text domain, often based on the following four clues: 1) frequency, 2) word association, 3) sentence/document structure or position, and 4) linguistic knowledge. TF-IDF weighting, as one of the simple yet robust frequency-based strategies, has been shown to be very effective for selecting important words for various text domains [4, 5, 6, 7, 8, 9]. It is based on the assumption that keywords should appear frequently in a specific document, but do not occur frequently in the entire document collection. Other frequency-related approaches use residual IDF, variance of term frequency, gain, burstness and so on [10]. The word association based approaches assume that important words tend to co-occur within a domain. Various knowledge resources have been leveraged to measure the association between word pairs, including the encyclopedia based lexical resources or a domain-specific thesaurus [11, 12, 13]. Web-based resources, such as Wikipedia or search engines, have also been

leveraged recently to determine the word associations [8, 14, 15]. Corpus-based methods, such as LSI (latent semantic indexing), MI (mutual information), and Chi-square statistic are also popular for computing word co-occurrence probabilities [16, 17]. Position-based approaches generally assume keywords are more likely to appear in special positions of the document, such as title, headline, first paragraph, or important sentences [6, 13, 18, 19, 20]. In particular, [13, 20] attempted to use an iterative reinforcement approach to do keyword extraction and summarization simultaneously, on the assumption that "important" sentences usually contain keywords, and keywords are usually seen in "important" sentences. Linguistic clues also play a significant role in locating important words, such as part-of-speech (POS) tag patterns [11, 21]. These information sources have been widely used in both unsupervised and supervised methods in previous studies. Even though unsupervised approaches have the advantage of domain independence and requiring no training data, a supervised framework can often better combine multiple knowledge sources and achieve strong discriminative power [4, 6, 16, 19, 22, 23].

Compared to the text domain, there have been very limited studies on speech data. [12] compared two lexical resources, WordNet and EDR electronic dictionary, for extracting keywords from multiparty meeting corpus. The systems or annotators were asked to select keywords from a list of simple nouns, which were pre-generated for each dialogue segment using a POS tagger. They showed that leveraging semantic resources can yield significant performance improvement compared to the approach based on the relative frequency ratio (similar to IDF). [15] attempted to eliminate mistranscribed keyphrases based on semantic coherence (measured using mutual information). They showed some positive results, however, sometimes correct keyphrases are also removed. [22] evaluated the performance of the tool "Extractor" on broadcast news transcripts with

various quality. They evaluated keywords extracted from automatic speech recognition output and compared them with those generated from reference transcripts.

2.2 Summarization

Automatic speech summarization research has adopted many techniques from text domain and showed that some of them generalize well to meeting domain. [24] applied a version of the Maximum Marginal Relevance (MMR) approach [25] to extract the most salient sentences from dialogue segments. The MMR approach is a linear model that combines salience and redundancy, retaining sentences that are most similar to the entire dialogue segment but are less similar to those sentences that are already selected in the summary. Other unsupervised approaches include the latent semantic analysis (LSA) approach [26, 27] and the concept-based integer linear programming (ILP) framework [28, 29]. The LSA approach first built a term-document matrix, then performed singular value decomposition (SVD) on it. Sentences with higher singular values are extracted to form the summary. The ILP framework used a constrained optimization setup to select sentences that maximize the coverage of a group of n-gram concepts while satisfying the length constraint (and thus inherently limit redundancy).

Graph-based summarization is another important line of work. In these approaches, sentences are often formulated as nodes in the graph, while lexical or semantic similarities between sentences represent edges. The similarity information is propagated or reinforced in each iteration. Sentences that are most similar to other sentences are selected to form the summary. Systems such as LexRank [30], TextRank [31], Manifold-ranking [32] have achieved satisfying performance in the written text domain. [33] proposed a ClusterRank approach to address the high redundancy

and low lexical density problem of the speech transcripts when applying the graph-based approach to the meeting domain.

Many speech summarization studies also utilize the supervised learning framework. Popular approaches include the hidden Markov model (HMM) [34], maximum entropy [35], conditional random fields (CRF) [36], Bayesian Network [37], and support vector machines (SVM) [38, 29]. A variety of lexical, structural, and acoustic/prosodic information (such as pitch, duration, energy, and pause) are utilized under the supervised framework for different speech domains [29, 36, 37, 38, 39, 40]. To better utilize speech-specific characteristics, [36] incorporated discourse cue words, listener feedback, and speaker activity related features in their meeting summarization system; [41] proposed to use re-occurring acoustic patterns in speech to estimate utterance similarity, hence identify salient utterances without using transcribed text; [42] investigated the hierarchical structure in lecture speech and developed a rhetorical state HMM for summarization; they also showed that speaker normalized acoustic features are highly effective for lecture summarization. [43] explored various ways to robustly represent the recognition hypotheses of spoken documents. Other than unsupervised and supervised approaches, [44] investigated semi-supervised meeting summarization using co-training algorithm to take advantage of two different views: the textual feature set and the prosodic/acounstic feature set. In [45], the authors developed a risk minimization framework that naturally combines supervised and unsupervised summarization models, and introduced various loss functions to measure the relationship between pairs of sentences.

As an attempt to generate abstractive speech summaries, [46] proposed an automatic speech summarization system that incorporates word significance measure, confidence measure, linguistic likelihood, and word concatenation probability to generate condensed broadcast news summaries.

[47, 48] developed a complete automatic abstractive summarizer that follows the interpretation-transformation-generation pipeline. The interpretation stage maps the sentences to a conversation ontology; the transformation stage selects the summary contents using an integer linear programming (ILP) approach; the final textual summary is generated through a surface realization process. To evaluate the usefulness of the generated summaries, [49] conducted a decision audit task using the meeting browser, where the users were asked to browse several meetings to understand how and why a decision was made, both keywords, extractive, and abstractive meeting summaries were provided to the users. They show that the gold standard abstractive summaries have a superiority in satisfying the user information need. Similar findings were also revealed in [48] where users prefer abstract-style summaries over extracts for browsing meeting transcripts.

Most recently, summarizing the user-generated contents (such as microblogs from the social network sites) has drawn increasing attention in the research community. In [50], the authors proposed a phrase reinforcement (PR) algorithm to automatically summarize any Twitter topics in one sentence. Given a topic phrase specified by the user and a set of Twitter posts containing this phrase, the algorithm first builds an acyclic graph using all the words in the posts and the topic phrase as the root node. Each word is represented by a node and weighted in proportion to the frequency of the phrase generated by following the path from the root node to this node. The summary sentence is selected as one of the highest weighted paths in the graph. [51] also proposed a hybrid TF-IDF approach that weights each sentence in the Twitter posts according to the average of the TF-IDF scores of the consisting words. While the term frequency (TF) was calculated by considering the entire collection of posts as a pseudo-document, the inverse document frequency (IDF) was calculated using single post as a document unit. The authors compared the hybrid

TF-IDF approach with the phrase reinforcement (PR) approach, as well as two naive baselines by randomly selecting posts or selecting the longest available post/sentence. Results show that the hybrid TF-IDF algorithm produces better summaries than other approaches according to the ROUGE-1 measure and a manually evaluated content score. Since both the PR algorithm and the hybrid TF-IDF approach were originally designed to generate short one-sentence summary, they can hardly cover all the information contents carried in the topic posts. [52] altered the hybrid TF-IDF approach to output four most highly weighted posts for each topic, and compared the results with a cluster-based summarizer, as well as the classic MEAD, LexRank, and TextRank systems. Experiments show that the TF-IDF based approach retained robust performance on the noisy Twitter posts and achieved better performance than the traditional text summarization systems. There are also studies working on visualizing the Twitter topics by identifying a set of topic phrases and presenting the related tweets to users [53, 54].

CHAPTER 3

MEETING CORPUS AND ANNOTATIONS

We perform our experiments on the ICSI meeting corpus, which will be briefly introduced in Section 3.1. The keyword and extractive meeting summarization annotations are presented in Section 3.2 and Section 3.3 respectively*, together with the inter-annotator agreement studies. Section 3.4 summarizes this chapter.

3.1 The ICSI Meeting Corpus

The ICSI meeting corpus [55] consists of 75 naturally-occurring meeting audio recordings, each about an hour long. These are mainly research discussions in the area of natural language processing, artificial intelligence, speech, and networking. All the meetings have been transcribed and annotated with dialogue acts (DAs) [56], topic boundaries, extractive and abstractive summaries [26] under the AMI project [57].

3.2 Keyword Annotation

We recruited two computer science students to annotate keywords and topic categories for each topic segment, using 26 meetings selected from the ICSI meeting corpus.† The annotators were

*Thanks to Feifan Liu for helping with the data annotation process.

†We selected these 26 meetings because they have been used in other previous studies for topic segmentation and summarization [26, 58]. Compared to the data set used in [59], one meeting (Bro015) is dropped because it is much

asked to listen to the audio recordings, read human transcripts, and select up to five words/phrases that can best convey the main content of a topic segment. Other than that, no specific annotation instructions were given. On average, it took one to two times real time for annotators to select keywords for each meeting. For topic labeling, six predefined high-level categories were provided based on the structural function of each topic segment, including "On-topic Discussion", "Digits", "Chitchat", "Opening", "Closing", and "Agenda". In the current experiments, we only use topic segments that are tagged as "On-topic Discussion" by both annotators, since these represent the main content of the meeting conversations. In total, there are 134 such topic segments for the 26 meetings. Note that keywords selected by human annotators are not restricted to single words. In fact, 66.06% of the total selected keywords are unigrams, 31.17% of them are bigrams, 2.25% are trigrams, and keywords with more than 3 words are very rare (0.52%).

In general, we found that annotators have quite high disagreement on keyword selection, similar to an observation in previous studies [60]. We use two metrics to quantify the inter-annotator agreement: Kappa coefficient [61] and consistency rate. Kappa measures the degree of inter-annotator agreement beyond the amount expected by chance. The Kappa score between the two annotators is 0.41 in our data. This moderate inter-annotator agreement seems reasonable considering the inherent difficulties due to the informal conversational style. Consistency rate, defined as the proportion of selected keywords that are agreed by the two annotators, is 26.71%. However, we also noticed that the consistency rate reaches 80% for several topic segments. This suggests that human annotation consistency may depend on different input. Some topic segments do not have a clear focus and the discussion in it is casual and lacks order; therefore it is very difficult to

shorter than all the other meetings.

decide what are the most representative keywords. In addition, some meetings contain technical discussions and are hard for annotators without the proper background to understand. In Table 3.1, we present the keywords annotated for the topic segment that corresponds to the sample meeting segment in Chapter 1.

Table 3.1. Human annotated keywords for the topic segment corresponding to the the sample meeting segment.

Example of human annotated keywords:	
Annotator 1:	interface, pedagogical belief-net, decision variables, m-three-l, x-schema
Annotator 2:	smartkom, belief-nets, combinatoric, decision, values

3.3 Extractive Summary Annotation

The ICSI meeting corpus has extractive and abstractive summaries [26] annotated under the AMI project [57]. To investigate the annotation consistency, we also perform our own extractive summarization annotation on the same 27 meetings as used for the keyword annotation. Three annotators (undergraduate students) were recruited to extract summary sentences on a topic basis using the topic segments from the AMI annotation. Each sentence corresponds to one DA annotated in the corpus. The annotators were told to use their own judgment to pick summary sentences that are informative and can preserve discussion flow. The recommended percentages for the selected summary sentences and words were set to 8.0% and 16.0% respectively. Human subjects were provided with the meeting audio files and an annotation graphical user interface, from which they can browse the manual transcripts and see the percentage of the currently selected summary sentences and words. We refer to the above annotation results on the 27 meetings as **Data set I**.

To examine the annotation consistency and investigate the factors that may have an impact on the inter-annotator agreement, we consider specifically the annotations on the 6 meetings used in [26], for which we have human annotated summaries using 3 different guidelines, as listed below. Both Kappa statistics and ROUGE scores are used to evaluate the human agreement in the extractive summarization annotation task.

- **Data set II**: summary annotated on a topic basis. This is a subset of the 27 annotated meetings above.
- **Data set III**: annotation is done for the entire meeting without topic segments.
- **Data set IV**: the extractive summaries are from the AMI annotation [26].

3.3.1 Kappa Statistic

Table 3.2 shows the average Kappa results, calculated for each meeting using the four data sets described above. Compared to Kappa score on text summarization, which is reported to be 0.38 by [62] on a set of TREC documents, the inter-annotator agreement on meeting corpus is lower. This is likely due to the difference between the meeting style and written text.

Table 3.2. Average Kappa scores on different data sets.

Data Set	I	II	III	IV
Avg-Kappa	0.261	0.245	0.335	0.290

There are several other observations from Table 3.2. First, comparing the results for Data Set (II) and (III), both containing six meetings, the agreement is higher for Data Set (III). Originally, we expected that by dividing the transcript into several topics, human subjects can focus better

on each topic discussed during the meeting. However, the result does not support this hypothesis. Moreover, the Kappa result of Data Set (III) also outperforms that of Data Set (IV). The latter data set is from the AMI annotation, where they utilized a different annotation scheme: the annotators were asked to extract dialog acts that are highly relevant to the given abstractive meeting summary. Contrary to our expectation, the Kappa score in this data set is still lower than that of Data Set (III), which used a direct sentence extraction scheme on the whole transcript. This suggests that even using the abstracts as a guidance, people still have a high variation in extracting summary sentences. We also calculated the pairwise Kappa score between annotations in different data sets. The inter-group Kappa score is much lower than those of the intragroup agreement, most likely due to the different annotation specifications used in different data sets.

We further analyze inter-annotator agreement with respect to two factors: topic length and meeting participants, using Data Set (I) with 27 meetings.

We computed Kappa statistic for each topic instead of the entire meeting. The distribution of Kappa score with respect to the topic length (measured using the number of DAs) is shown in Figure 3.1. When the topic length is less than 100, Kappa scores vary greatly, from -0.065 to 1. Among the entire range of different topic lengths, there seems no obvious relationship between the Kappa score and the topic length (a regression from the data points does not suggest a fit with an interpretable trend).

Using the same Kappa score for each topic, we also investigated its relationship with the number of speakers in that topic. Here we focused on the topic segments longer than a threshold (with more than 60 DAs) as there seems to be a wide range of Kappa results when the topic is short (in Figure 3.1). Table 3.3 shows the average Kappa score for these long topics, using the number of

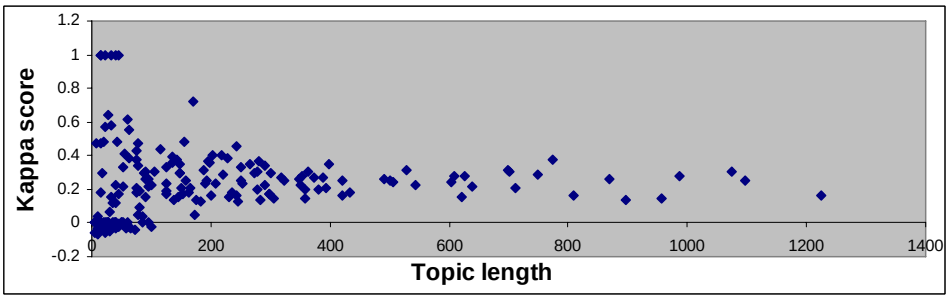


Figure 3.1. Relationship between Kappa score and topic length.

speakers in the topic as the variable. We notice that when the speaker number varies from 4 to 7, kappa scores gradually decrease with the increasing of speaker numbers. This phenomenon is consistent with our intuition. Generally the more participants are involved in a conversation, the more discussions can take place. Human annotators feel more ambiguity in selecting summary sentences for the discussion part. The pattern does not hold for other speaker numbers, namely, 2, 3, and 8. This might be due to a lack of enough data points, and we will further analyze this in the future research.

Table 3.3. Average Kappa score with respect to the number of speakers after removing short topics.

# of speakers	# of topics	Avg Kappa score
2	2	0.204
3	6	0.182
4	26	0.29
5	26	0.249
6	33	0.226
7	19	0.221
8	7	0.3

3.3.2 Agreement Measured Using ROUGE

ROUGE [63] has been adopted as a standard evaluation metric in various summarization tasks. It is computed based on the n-gram overlap between a summary and a set of reference summaries. Though the Kappa statistics can measure human agreement on sentence selection, it does not account for the fact that different annotators choose different sentences that are similar in content. ROUGE measures the word match and thus can compensate this problem of Kappa.

Table 3.4 shows the ROUGE-2 and ROUGE-SU4 F-measure results. For each annotator, we computed ROUGE scores using other annotators' summaries as references. For Data Set (I), we present results for each annotator, since one of our goals is to evaluate the quality of different annotator's summary annotation. The low ROUGE scores suggest the large variation among human annotations. We can see from the table that annotator 1 has the lowest ROUGE score and thus lowest agreement with the other two annotators in Data Set (I). The ROUGE score for Data Set (III) is higher than the others. This is consistent with the result using Kappa statistic: the more sentences two summaries have in common, the more overlapped n-grams they tend to share.

Table 3.4. ROUGE F-measure scores for different data sets.

		ROUGE-2	ROUGE-SU4
data (I)	Annotator 1	0.407	0.457
	Annotator 2	0.421	0.471
	Annotator 3	0.433	0.483
data (III)	2 annotators	0.532	0.564
data (IV)	3 annotators	0.447	0.484

3.4 Summary

In this chapter, we briefly introduce the ICSI meeting corpus and the data annotation scheme for both keyword extraction and extractive summarization tasks. Inter-annotator agreement for these annotations was measured using various evaluation metrics. Compared to the keyword and summary annotations conducted on the written text domain, human agreement on meeting transcripts was much lower. This could be due to the informal style of meeting conversations, their lack of structure information, the ill-formed speech utterances and redundant information contained in them, etc. In addition, the specialized topics in the ICSI corpus could be difficult for the annotators without proper background knowledge (we recruited undergraduate students to conduct all annotations). In [64], we propose to use divergence distance scores to evaluate the annotation quality for each of the annotators, and delete potentially incoherent summary sentences.

CHAPTER 4

KEYWORD EXTRACTION *

In this study, our task is to extract keywords for each of the topic segments in the meeting transcript. Therefore we will use "topic segment" and "document" interchangeably in the following of the chapter, to represent the individual processing unit. In all the experiments, we consider lemmatized content words (i.e., noun, verb, adjective and adverb) as keyword candidates. The core part of keyword extraction is to assign a salience score to each word, such that the system selects top ranked words as keywords. In Section 4.1, we present the unsupervised TF-IDF weighting as the baseline approach. A supervised framework with single-loop feedback strategy is introduced in Section 4.2, followed by the experimental results and analysis in Section 4.3. Finally, we conclude the chapter and present discussions in Section 4.4.

4.1 Unsupervised TF-IDF Weighting

We used the TF-IDF framework as our baseline, since this approach has been shown to be very effective in previous keyword extraction studies. Under this framework, candidate words with the highest TF-IDF scores will be selected as keywords. The term frequency (TF) for a word w_i in a

document is the number of times the word occurs in the document. The IDF value is:

$$IDF_i = log(N/N_i)$$

where N_i denotes the number of documents that contain word w_i, and N is the total number of documents in the collection. We use the meeting corpus as the document collection for IDF calculation.

4.2 Supervised Framework

Under the supervised framework, each candidate word is represented by a feature vector. We use a maximum entropy classifier to assign the posterior probability of a word being a keyword. Section 4.2.1 provides details of the features we use in the supervised classification framework. Section 4.2.2 explains the feedback strategy we propose to reinforce the effect of important sentences on the task of keyword extraction.

4.2.1 Features

We first describe several widely used features in prior work for keyword extraction, and then list the features we propose.

- Commonly used features:
  - We use three frequency related features: TF (term frequency), IDF (inverse document frequency), and the product of them, TF-IDF. These effectively identify words that appear frequently in a document, but do not occur frequently in the entire document collection.

  - Position features are used to represent the first occurrence of a candidate word. We compute this on a sentence or word basis, named 'dist-sent' and 'dist-word' respectively.
  - Sentence length and salience score features are extracted from the sentences containing the word. For a sentence, its length is represented by the number of word tokens it contains, and its salience score is calculated based on its cosine similarity with the entire meeting under the vector space model. If a candidate word appears in several sentences, we use the length of the longest sentence and the highest sentence salience score among those sentences.[†]

- Term specificity features:

  Term specificity is generally defined as "the extent to which the word's referent can be touched or felt" [65]. Terms with high specificity usually carry more semantic content. As an example, "chipmunk" is considered a more specific term than "animal". Intuitively, these words with high specificity should be weighted more heavily in tasks such as information retrieval, since these words play a significant role in characterizing the document [10]. By contrast, stopwords contribute the least amount of information content, and should be removed or downweighted. Based on this motivation, we introduce two types of features for term specificity.

  The first feature is the number of senses a word has. Since specific words generally have more precise meaning, it is natural that they have fewer senses. This negative correlation

[†]We also experimented using the average length and salience score of the sentences, but these did not yield better performance.

between term specificity and the number of word senses has been confirmed in [10]. In this work, we find the number of word senses from WordNet (version 1.7.1).

Another feature we use to represent term specificity is based on stopwords. Since a word with low IDF score means that it occurs in many documents and is not topic indicative, we create stopword lists consisting of words with the lowest IDF values. Three binary features are defined: 'sw-200', 'sw-300', and 'sw-500', to denote whether a candidate word is in the corresponding list with 200, 300, and 500 stopwords, respectively.

- Decision-making (DM) sentence features:

  We notice that some human selected keywords are likely to appear in decision making (DM) sentences. Similar findings have been confirmed in [66, 67]. According to [66], decision making conversations are more likely to contain the indicative word *we*, such as "we should", "we will"; thus we extract those sentences with a structure of "we + any verb + any noun" and consider them as an approximated collection of decision-making sentences. A binary feature 'DM-in' is defined to indicate whether the target word has appeared in the DM sentences. In addition, we found that some adverbs (such as "actually", "basically", "especially") are also commonly used in conversations to emphasize a sentence or clarify a point; therefore, we use a binary feature 'DM-adv' to indicate the co-occurrence of a target word and some predefined adverbs that are collected from the entire meeting corpus (after removing some frequent adverbs since they have very weak discriminative power).

- Summary features:

  Given a summary, we extract four features from it: a binary feature indicating whether a

candidate word has appeared in the summary (summary-in); the frequency of the word in the summary (summary-tf); the normalized frequency by its total number of occurrences (tf-norm); as well as the ratio of its occurrence in summary sentences and non-summary sentences (tf-ratio). We notice that these features favor different summary lengths in order to have more discriminative power, for example, shorter summaries by the first feature (summary-in) and longer summaries by the other three features; therefore, we use summaries of different length for these features. We used system summaries that contain about 20%, 30% and 40% of the total word tokens. These percentages are selected empirically with the expectation that they can cover most of the salient sentences in the document and provide discriminative information for keyword extraction. In addition, these are in accordance with the typical compression rate range used in summarization [68].

One simple method to generate a system summary is to select the longest sentences in the document until reaching the predefined compression ratio. We name this method "TopLen" and use it as a baseline. This approach has been shown to be a very strong baseline for speech summarization [69]. In Section 4.2.2, we will investigate other query-focused summary generation approaches.

- Speech related features:

  Even though our task is to extract keywords from transcripts, we do have access to the speech signal and speaker information (the corpus we used contains recordings from separate channels for different speakers). We use two features specific to conversational speech.

  The first one is related to speakers. In meeting conversations, important words may be said by many participants. We therefore define the feature 'spkr-num' as the number of speakers

who have mentioned the candidate word.

The second feature is based on prosody (how a word is said). We expect that words with high pitch accent (i.e., prominence) are more likely to be keywords. In this work, a word-level pitch accent detection module trained using read speech [70] is employed to generate the pitch accent likelihood score for each candidate word.[‡] For a word that appears more than once, we take the average of the likelihood scores associated with each appearance. We refer to this feature as 'prominence'.

- POS features:

  For a candidate word w with POS tag t_i, a feature '$pos(t_i)$' is defined as the relative frequency of this word with tag t_i within the document. In addition, we hypothesize that the POS of words around a candidate word may be useful cues for determining keywords. Therefore, we include similar tag frequency features for the previous and the following word, denoted as '$pos\_before(t_i)$' and '$pos\_after(t_i)$'.

4.2.2 Single-Loop Feedback Strategy

The relationship between summarization and keyword extraction has recently been investigated in written text domain [13, 20, 71]. However, the graph-based mutual reinforcement approach seems to underperform on the meeting transcripts, as shown in [72]. This is partly due to the informal conversational style. Therefore, we propose a single-loop feedback strategy to generate more keyword-related sentences while maintaining the supervised framework. We first use the

[‡]Thanks to Je Hun Jeon for generating the pitch accent scores.

supervised method with only a few base features to generate keywords, then employ the query-focused sentence retrieval approach [31, 73, 74] to generate summaries. These summaries are further used to extract summary-related features described above. This procedure is illustrated in Figure 4.1.

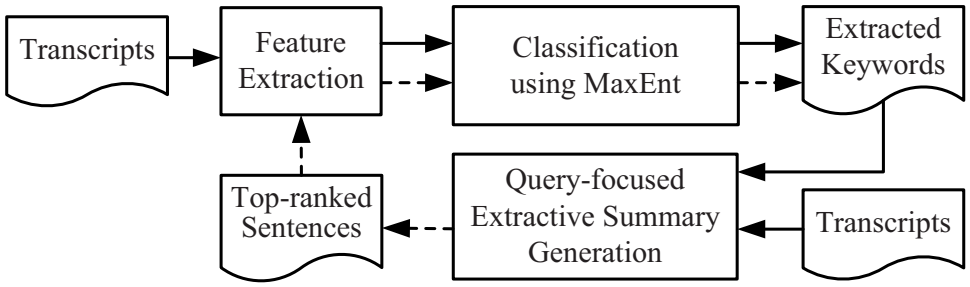


Figure 4.1. Single-loop feedback strategy for keyword extraction.

The upper part of the figure shows the supervised keyword extraction process described above, with data flow along the solid lines and arrows. This approach uses a set of base features, including TF, IDF, TF-IDF, part-of-speech, and stopwords features. The lower part of the figure illustrates the feedback loop using query-focused summaries to extract features for another round of keyword extraction. The top $h\%$ (we use 60%) of words with highest confidence scores from the first pass keyword extraction are used as input query words, and all the sentences are ranked using a query-focused approach [31, 73]. The score of a sentence s given a query q, $p(s|q)$, is calculated as the weighted sum of the sentence's similarity to the query and its similarities with the other sentences in the document, along with those sentences' salience scores. Sentences' salience scores are calculated in an iterative process. For the $(k+1)^{th}$ iteration,

$$
\begin{aligned}
p^{[k+1]}(s|q) &= d\frac{sim(s,q)}{\sum_{z\in C} sim(z,q)} \\
&+ (1-d)\sum_{v\in C}\frac{sim(s,v)}{\sum_{z\in C} sim(z,v)}p^{[k]}(v|q)
\end{aligned}
$$

where $sim(\cdot,\cdot)$ is the similarity measure between two sentences or a sentence and a query, C is the set of all sentences in the document, and d is a trade-off parameter to weight the contribution from each component. A larger d means the sentence's relevance to the query is weighted higher. We set d as 0.6 in our experiments.

We employ the vector space model to calculate the sentence-sentence and sentence-query similarity, that is, each sentence and the query are represented as a vector of terms, then we calculate dot product between the two vectors. This approach favors longer sentences since previous research has shown that long sentences are very informative and can form appropriate summaries [69]. We consider two approaches to assign term weights when building the vectors for the sentences, and name the resulting summaries accordingly:

- TFIDF: product of a word's term frequency in the document and its inverse document frequency
- CONF: confidence score from the supervised classifier in the first pass

We further explored a simple reranking process to achieve maximum diversity in the top ranked sentences with respect to the contained query words. The basic idea is to push a sentence higher if it contains query words that are not seen before, and conversely lower if a sentence does not have any new query words. The reranking approach is shown in Algorithm 1. Its effect on diversifying the top-ranked sentences will be evaluated in Section 4.3.4.

Algorithm 1 Algorithm for reranking the summary sentences

Let S be the current ranked summary sentences
Let $N = |S|$ be the number of summary sentences in S
Let $R = \phi$ be the current ranked list of sentences
Let $Q = \phi$ be the set of query words
for i from 1 to N **do**
  if sentence s_i contains query words that are not in Q **then**
    add s_i to R and remove it from S
    add the new query words to Q
  end if
end for
Append S to R
Output R (reranked summary sentences)

4.3 Experiments

4.3.1 Experimental Setup

We evaluate automatic keyword extraction performance using different testing transcripts. In addition to human transcripts, we use different ASR transcripts obtained from a state-of-the-art SRI recognizer [75, 76]. The first one is the final recognition output after rescoring. It has a moderate word error rate (WER) of about 36.2% on our corpus. For this condition, we obtained the DA and topic boundary information by aligning the human annotation to the ASR words. We will refer to this condition 'ASR' when there is no ambiguity (compared to its general meaning of automatic speech recognition). The second is n-best hypotheses from the recognizer. We will use both the 1-best and additional candidates in our experiments. The WER for the 1-best is higher than the above ASR transcript, about 41.6% for the 26 meetings. For this condition, we used the speech recognizer's segments, which are typically pause based, therefore each resulting transcript segment does not correspond to a DA. These different testing scenarios are used to examine the effect of recognition and sentence segmentation errors on keyword extraction.

Table 4.1. Statistics about the 26-meeting corpus. The numbers are generated and averaged across all 134 topic segments. "S.D." stands for standard deviation.

Statistics of corpus	Human	ASR	1-best
Avg. num of words	1,867	1,704	1,861
Avg. num of sentences	269	219	202
Avg. sentence length	6.94	7.77	9.20
S.D. of sentence length	7.91	7.55	12.48
Avg. num of annotated keywords	5.92	-	-
S.D. of annotated keywords	2.18	-	-
Percentage of covered keywords (%)	100	65.48	65.00
Avg. WER (%)	0.00	36.19	41.62

Table 4.1 shows some statistics about the corpus used in this study. These scores are generated and averaged over all the topic segments. The percentage of the covered keywords measures how many of the human annotated keywords appear in the recognition output. As mentioned before, since the segmentation for 1-best was pause-based, the average "sentence" length is generally longer for that condition than the others (human transcripts or the ASR output with DA boundaries aligned from the reference ones), and the variance is also much larger. We also notice that, although "1-best" output has higher WER than "ASR", the keyword coverage rate is about the same for the two conditions.

For all the transcripts (human or recognition output), we used the TreeTagger [77] to lemmatize them, and the TnT part-of-speech tagger [78] trained from the Switchboard data to tag the transcripts. For the supervised framework, we use the maximum entropy classifier [79], and perform 9-fold cross validation on the 26-meeting corpus in all the experiments.

For a comparison, we adopt the state-of-the-art keyphrase extraction system "Kea"[§], which has been shown to perform satisfyingly on a variety of tasks. Kea uses Naive Bayes learning algorithm

[§]http://www.nzdl.org/Kea/

with four basic features: TF-IDF, first occurrence, length of the phrase, and node degree of the candidate phrase. We use the default setting of the Kea system without any controlled vocabulary.

Different performance metrics were employed in [72], including both automatic evaluation and human evaluation. Since in general human evaluation results are correlated with the automatic metrics, we choose not to conduct human evaluation. For automatic evaluation, we use widely adopted precision/recall/F-score measurement. Given a set of system hypothesized unigrams and the corresponding human reference keywords, precision P is calculated as the number of matched unigrams, divided by the total number of system hypotheses; while recall R is the number of matched unigrams divided by the number of human reference words. The F-score is computed as the harmonic mean of precision and recall:

$$F_1 = \frac{2 \times P \times R}{P + R}$$

We use each annotation as a reference, and then compute the average score as the final result. Another automatic evaluation metric used in [72] is the weighted relative score that considers references from multiple annotators together and according to that, weights words differently. Because we use only two annotations in this study, this evaluation is not substantially different from a simple average of the $P/R/F_1$ scores above, therefore, we only use one metric in this work. We perform the evaluation on a lenient unigram basis, that is, both the system hypotheses and human annotated keywords are first lemmatized, then compared to each other on a unigram basis.

4.3.2 Results

Table 4.2 shows the results of our supervised keyword extraction approaches, in comparison with the unsupervised TF-IDF weighting and the Kea system. We performed significance test using

McNemar test for the F-scores with respect to the TF-IDF baseline system. For each method, we output 5 unigrams in this experiment. For the supervised approach, we show results for three different settings according to the feedback module and summary generation approaches:

- **Supervised-TopLen:** Without feedback; summary-related features are extracted from "TopLen" summaries.
- **Supervised-TFIDF:** With feedback; summary-related features are extracted from "TFIDF" summaries.
- **Supervised-CONF:** With feedback; summary-related features are extracted from "CONF" summaries.

Table 4.2. Keyword extraction results for both human transcripts and ASR output using TF-IDF/Kea/supervised approaches.
$\star$ and † mean that the improvement of the supervised approach over the TF-IDF method is statistical significant at the confidence level of 95% and 90% respectively.

		P(%)	R(%)	F(%)
Human	TF-IDF	35.33	32.50	33.78
	Kea	36.14	33.19	34.49
	Supervised-TopLen	39.02	35.38	37.01*
	Supervised-TFIDF	41.59	37.10	39.11*
	Supervised-CONF	**42.19**	**38.28**	**40.03***
ASR	TF-IDF	26.12	23.71	24.81
	Kea	27.95	25.34	26.50
	Supervised-TopLen	27.40	25.86	26.52
	Supervised-TFIDF	29.14	27.04	27.96†
	Supervised-CONF	**29.44**	**27.93**	**28.56**†

As can be seen from Table 4.2, based on the F-score, the Kea system outperforms TF-IDF weighting by 0.71% and 1.65% (absolute) on human transcripts and ASR output, respectively.

This indicates that the keyword extraction approaches developed in written text domain may not be directly applied well to the conversational speech data. By contrast, our supervised approach with "CONF" summaries outperforms TF-IDF weighting by 6.25% (abs.) on human transcripts, and 3.75% (abs.) on ASR output. The improvement is observed consistently across both recall and precision rates. When compared to the "Supervised-TopLen" system, which uses the same feature sets except the summary related features, the "Supervised-CONF" system yields an improvement of about 3% and 2% respectively on human and ASR transcripts. This shows that the summary features generated after the feedback loop are effective, boosting keyword extraction performance. More detailed analysis about using summaries for keyword extraction is provided in Section 4.3.4.

In Table 4.3, we compare the human annotated keywords with system generated hypotheses for both human transcripts and ASR output. The system hypotheses are generated using the supervised system with 'CONF' weighting for summary generation. The lemmatized human annotated keywords are provided for comparison. On human transcripts, our system successfully extracted "action", "intention", "domain" and "rad" as keywords, while on ASR output only "action" and "intention" were extracted. This is partly because as many as 34.52% of the human annotated keywords are missing in the ASR output due to the recognition errors. For example, "domain", "parser input", "binding" were recognized as "main", "part be towards", and "finding" respectively. This high rate of missing the reference keywords results in an upper bound of recall rate (around 61.5% in our data) — no matter what the keyword extraction approach is, these reference words will not be hypothesized in the system output. This example shows that ASR errors can significantly impact keyword extraction performance, resulting in misses of some important words, change of the weight of desired keywords, and maybe selection of incorrectly recognized words. We will make

further investigation about the effect of n-best ASR output on keyword extraction in Section 4.3.5.

Table 4.3. Comparison between human annotated keywords and system generated hypotheses. All the keywords are lemmatized.

Annotator 1	action intention binding
Annotator 2	intention domain object rad binding parser input
System (Human)	action intention domain rad schemas
System (ASR)	action intention specific module general

4.3.3 Analysis I: Feature Analysis

To measure the effectiveness of different features, we perform three feature selection processes.

- **Forward feature selection.** It begins with an empty set, and iteratively adds the feature that achieves the largest performance gain when combined with the current selected features.
- **Backward feature selection.** It starts with the entire feature set, and removes one feature in each iteration to have the least performance degradation.
- **Dynamic programming (DP) based feature selection.** Similar to forward feature selection, this approach also starts with an empty set. Unlike forward and backward methods that only keep the best feature subset in each iteration, this approach maintains N best feature subsets from a set of N features in each iteration using a dynamic programming based strategy [80].

In all three processes, the final feature set is the one that achieves the best performance during the iterations. The three approaches have their own merits, though none of them can guarantee the selection optimality. Forward and backward feature selection approaches adopt the greedy strategy

but they are computationally more efficient; dynamic programming based approach enlarges the searching space and results in feature subsets with more divergence.

Since our annotated data is limited, we perform this feature analysis using all the data in the same cross-validation setup. When more data is available, we would like to test the selected feature set on a blind test set. From this experiment, we hope to exploit the best performed feature combinations on the conversational speech data for the keyword extraction task.

Table 4.4. Feature selection results using forward (noted as 'Fw'), backward (noted as 'Bw'), and dynamic programming approaches (noted as 'Dp'). 'X' indicates the corresponding feature is selected. '-' means the feature is not available. The last row shows the F-score using the selected best feature subset.

		Human			**ASR**		
Category	**Features**	Fw	Bw	Dp	Fw	Bw	Dp
Frequency	TF	X	X	X		X	X
	IDF		X	X	X		X
	TF-IDF	X	X	X	X	X	X
Position	dist-word	X	X	X	X		X
	dist-sent					X	X
Sentence	sent-len		X				
	sent-score	X	X	X	X	X	X
Term specificity	sense-num	X	X	X	X	X	X
	sw-200	X	X		X	X	X
	sw-300	X	X	X			
	sw-500			X			X
DM-related	DM-in		X	X	X	X	
	DM-adv	X	X	X			
Summary	summary-in	X	X	X	X	X	X
	summary-tf	X	X	X	X	X	X
	tf-norm	X	X	X	X	X	X
	tf-ratio	X	X	X	X	X	X
Speech	spkr-num			X		X	
	prominence	X	X		-	-	-
POS	pos	X		X	X	X	X
	pos-context		X	X		X	
F-score (%)		41.22	41.38	42.34	29.93	30.10	30.52

Results are shown in Table 4.4. Not surprisingly, the traditional 'TF-IDF', 'pos', 'sent-score' features all perform well on the speech data. In addition, the linguistically motivated term specificity feature, such as 'sense-num', and the summary-related features also play a significant role in boosting the performance. By contrast, features 'dist-sent' and 'sent-len' are rarely selected in the final set. This could be because the spoken sentences are usually poorly structured and there is a large variation in length, as shown in Table 4.1. The speech-related 'prominence' feature also benefits the keyword extraction task on human transcripts. We did not include this feature for ASR condition, as the word alignment information was not available. It is also worth pointing out that since the prominence prosody model was trained using read speech, this may have limited its benefit on conversational speech due to the mismatched conditions. We will continue to investigate using prominence information in the future.

Overall, we observe improved performance after the feature selection processes, compared to using all the features. Among the three feature selection approaches, dynamic programming based approach achieves slightly higher performance; while there is no significant performance difference between the forward and backward feature selection processes. In addition, there are many common features in the final selected features from the three approaches. We also conducted an oracle experiment using human reference summaries [64] to generate summary-related features, and achieved an F-score of 43.30% and 43.18% respectively for forward and backward processes on human transcripts. This better performance over using system-generated summaries indicates that further improvements using better summaries are still possible for keyword extraction.

4.3.4 Analysis II: Impact of Summaries on Keyword Extraction

Although the relationship between keyword extraction and summarization has received some attention in recent years, limited research has been conducted on what kind of summary can boost keyword extraction performance. As shown in Table 4.2, different summaries can result in very different keyword extraction performance. We have seen that the confidence score based "CONF" summaries perform better than the "TopLen" and "TFIDF" summaries on both human transcripts and ASR output. In this section, we will analyze the effect of different summaries, expecting this line of work can benefit further research.

First, we evaluate if the reranking procedure (see Section 4.2.2) helps generate more informative summaries for keyword extraction. Table 4.5 shows the performance of different supervised systems. "NoR" means the corresponding summary is generated via the feedback mechanism but without the reranking process. We can see that the supervised "CONF" system achieves the best performance with the reranking module, with an improvement of F-score from 37.20% to 40.03% on human transcripts, and 26.86% to 28.56% on the ASR output, compared to without using reranking. For "TFIDF", the improvement from reranking is not as large as in "CONF", especially for the ASR condition. In the same table, we also show the keyword coverage statistics of different summaries. After reranking, "CONF" summary contains more than 10% of the human annotated keywords compared to the "TopLen" summary. This holds for different summary lengths and on both human and ASR transcripts. The better keyword coverage in the summaries partly explains the better performance using the corresponding summaries for keyword extraction.

For further analysis, we evaluate the correlation between the summary quality and its impact on keyword extraction. In total, we define seven measurements. Four keyword-related statistics

Table 4.5. Keyword extraction performance using features from different summaries. Also shown is the coverage of reference keywords in different summaries, with length about 20%, 30% and 40% of the total word tokens. "NoR" means without the reranking process.

		Extraction	Keyword Coverage %		
	Summary	F-measure %	Len. 20%	30%	40%
Human	TopLen	37.01	59.31	73.27	81.94
	TFIDF-NoR	37.63	68.46	77.21	83.95
	CONF-NoR	37.20	67.82	79.86	87.32
	TFIDF	**39.11**	**70.95**	**80.82**	**88.76**
	CONF	**40.03**	**71.67**	**83.87**	**91.33**
ASR	TopLen	26.52	34.03	42.86	49.60
	TFIDF-NoR	27.68	45.67	52.09	56.26
	CONF-NoR	26.86	44.70	51.52	56.58
	TFIDF	**27.96**	**47.59**	**53.21**	**58.51**
	CONF	**28.56**	**45.83**	**53.61**	**59.31**

are calculated to measure the distribution of keywords in the summaries:

1. Keyword Coverage (ST_1): percentage of reference keyword types that are covered in summary. This is the statistic shown before.

2. Normalized Keyword Frequency (ST_2): total frequency of reference keywords contained in the summary divided by the total frequency of reference keywords in the entire transcript.

3. Keyword Percentage (ST_3): total number of reference keywords contained in the summary divided by the total number of word tokens in the summary.

4. Sentence Percentage (ST_4): number of sentences that contain at least one reference keyword, divided by the total number of sentences in the summary.

Another three measurements come from the ROUGE scores [63], which calculate the word overlap between a system summary and the human reference summary. We use the reference

summary that is described in [64]. In total, we generated three ROUGE scores:

5. ROUGE-1: unigram match.

6. ROUGE-2: bigram-based match.

7. ROUGE-SU4: skip-bigram plus unigram-based match.

For the three types of summaries ('TopLen', 'TFIDF', and 'CONF'), we used different length from 10% to 50% of the total word tokens, with 5% interval. The unsupervised TF-IDF weighting was used to perform keyword extraction using these summary sentences with different compression ratios. This approach is the same as that used for the entire document, except that only a subset of the original sentences are used as input in this experiment. We then compute the correlation (Spearman's rho) between the F-scores and the different measurements described above. The correlation results are shown in Table 4.6 for both human transcripts and ASR output.

Table 4.6. Correlation between keyword extraction F-scores and various statistics for summaries.

	Human	ASR
ST-1	**0.7808**	**0.8315**
ST-2	**0.8742**	**0.8938**
ST-3	0.3999	0.4084
ST-4	-0.0116	-0.2821
ROUGE-1	-0.2320	-0.0049
ROUGE-2	0.5037	0.7796
ROUGE-SU4	0.2173	0.5611

From the correlation results, we observe that Normalized Keyword Frequency (ST_2) is highly correlated with keyword extraction performance, for both human transcripts and ASR output. Keyword Coverage (ST_1) measure is the second most correlated one. This is also consistent with

our findings in the feature selection results, where the summary-related features 'summary-in', 'summary-tf' and 'tf-norm' features are all included in the final feature combinations. In contrast, among the three ROUGE scores, only ROUGE-2 shows reasonable correlation with the F-score on ASR output, while ROUGE-1 and ROUGE-SU4 scores have lower correlation with the keyword extraction performance, suggesting that the criteria used to optimize summarization and keyword extraction are different: ROUGE measures the matches of all the words in the summaries, whereas for keyword extraction, it is more important to ensure a good coverage of keywords. Overall it seems that both ST_1 and ST_2 statistics can be used as reasonable indicators of whether the generated summaries are useful for the keyword extraction task.

For this analysis, we used unsupervised keyword extraction. We made this choice for different reasons. The unsupervised method is much simpler computationally and its performance is reasonably good. Also, the unsupervised approach can use different summary lengths and thus creates more data points for statistical analysis (e.g., 9 results for each of the summary types, corresponding to compression ratios ranging from 10% to 50% with 5% interval). In contrast, the supervised approach uses the summaries with different length together to extract features and provides one result corresponding to one summary type. Even though the unsupervised performance is not as good as the supervised approach, the general trend is similar between the two methods in terms of the effectiveness of the summary sentences (we performed analysis using supervised keyword extraction and observed similar patterns). Note that this correlation study is only for analysis purpose. We cannot use these measurements during keyword extraction since the reference keywords are unknown beforehand.

4.3.5 Using N-best Hypotheses for Keyword Extraction

As we have seen, there is a degradation when using ASR output for keyword extraction, mainly due to the word errors. In this section, we investigate if using rich ASR output, e.g., n-best hypotheses, can help improve performance. We use n-best hypotheses for $n = 1, 2, ..., 10, 15, 20$ in this experiment. Since our supervised framework requires the generation of different summaries, it is not straightforward to use it on n-best ($n > 1$) output (summarization using n-best is still an area that is understudied). Hence, we applied two keyword extraction systems, the unsupervised TF-IDF weighting and the Kea system. For both methods, we put n-best hypotheses together for each sentence, and then apply the algorithm as is to the expanded transcripts for each document. The keyword extraction results are shown in Figure 4.2 (left Y-axis). For each n-best, we also calculate its coverage of reference keywords, shown in the same figure (right Y-axis).

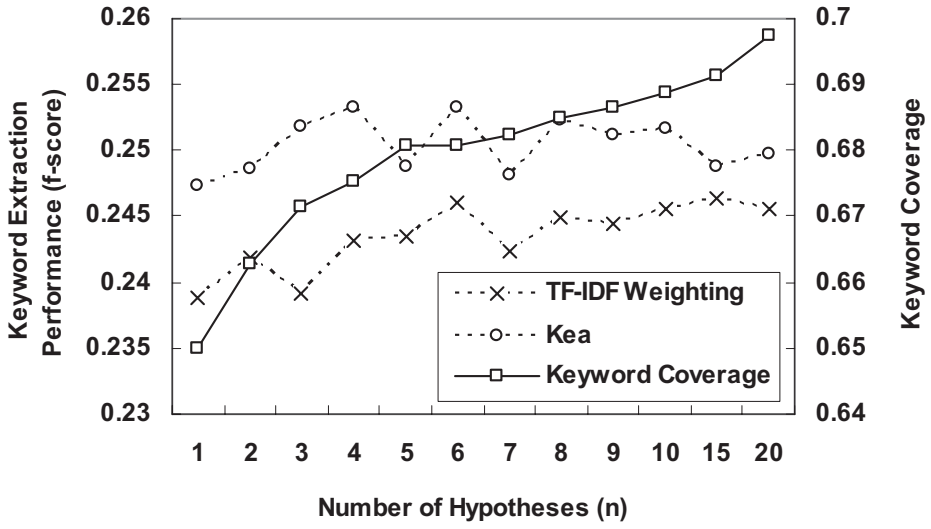


Figure 4.2. Keyword extraction performance (left Y-axis) and reference keyword coverage (right Y-axis) using n-best output.

We can see from the figure that the keyword coverage gradually improves as the number of hypotheses increases, compared to the coverage of 1-best result (65%). For both keyword extraction systems, there is a general trend of improved performance when n increases from 1 up to 6, even though there is some fluctuation as n changes. When n is 6, both TF-IDF weighting and Kea system achieve the best performance. After that, although the keyword coverage continues to increase, there is no improvement in keyword extraction performance. This preliminary study on n-best hypotheses shows that, more ASR hypotheses may contain more reference keywords and result in better keyword extraction performance. However, too many hypotheses may also introduce confusion to the keyword extraction task. Similar findings have been shown in [22]. Therefore, how to effectively utilize rich ASR output (n-best or lattices) is still a challenging problem for keyword extraction on speech transcripts.

4.4 Summary and Discussions

In this chapter, we focus on extracting keywords from meeting transcripts using a supervised framework with single-loop feedback strategy. We investigate a variety of novel features under the supervised framework, including linguistically motivated term specificity features, decision-making sentence related features, prosodic prominence scores, as well as a group of features derived from summary sentences, and conduct extensive analysis to demonstrate the effectiveness of the newly proposed features and the feedback mechanism used to generate summaries. Furthermore, we show promising results using n-best recognition output to address the problems of recognition errors.

In below, we summarize a few findings regarding keyword extraction from speech transcripts

based on our error analysis. We hope these different error categories will point out some future research directions.

- **Frequency-based methods.** Most of the approaches, whether it is unsupervised TF-IDF weighting or the supervised approach as used in this study, rely heavily on a word's frequency in the document to determine its importance. However, human annotated keywords may not occur frequently in the transcripts. In fact, we found that about 35% of the human annotated keywords occur only once or twice, and about 57% of the annotated keywords occur less than 5 times in the corresponding meeting transcripts. For example, "hire" was selected as a keyword by one annotator for a topic segment, because the meeting participants discussed to hire a student worker. However, "hire" was only mentioned once since the conversation later shifted to how to put the student on the payroll. A more intelligent system needs to understand the semantic relatedness to identify the low-frequency keywords.

- **Human annotation agreement and evaluation issues.** Annotating keywords is considered difficult by our annotators, and the human annotation agreement is not high for meetings. This poses problems for both training classification models and evaluating system performance. Furthermore, since human annotators only mark a fixed number of keywords, a system may generate a keyword that is not on the list of human annotated keywords but still acceptable. Hence, the current evaluation metric may be too strict. Human evaluation was used in [72] where human judges were asked to reject unacceptable keywords. That resulted in higher keyword extraction scores. However, human evaluation is expensive and not always feasible for system development. How to define a more effective annotation and evaluation scheme for speech transcripts therefore remains to be an interesting topic.

- **ASR errors.** High WER explains some errors in keyword extraction. Compared to F-measure of 40.05% using human transcripts, we obtain an F-score of 28.56% on ASR output (WER is 36.19%), and 26.22% on 1-best output (with WER of 41.62%), all using the "Supervised-CONF" system. The word errors cause serious problems especially when the reference keywords are incorrectly recognized. In addition, they have a negative impact on various features, such as POS tagging, parsing, semantic relatedness, and as a consequence degrade keyword extraction performance. Our preliminary investigation using n-best hypotheses has shown some promising results. Therefore, effectively using the n-best hypotheses or lattices, such as taking into account the confidence measures [44, 81], may compensate for the high word errors and is an important direction for developing keyword extraction systems for spoken documents.

- **Unigram-based approaches.** The current methods we use are based on single words. As described in Chapter 3, a large percent of the human annotated keywords are phrases. Since we only generate 5 unigram words, there will be missing keywords. In [59], a web resource based approach was used to generate bigrams. In a preliminary study, we also noticed that some top ranked unigrams can be naturally combined into bigrams or trigrams. After this combination, we can then generate additional unigram keyword candidates. In this way, the system output contains both unigrams and phrases, with a total of more than 5 words. This yields a higher recall rate, without much loss of precision, and thus better F-measure. We will investigate along this direction and try to extract more robust keyphrases in our future work.

CHAPTER 5

EXTRACTIVE MEETING SUMMARIZATION

In this chapter, we investigate the extractive meeting summarization task, where important dialogue acts (DAs) in the transcripts are selected to form a summary according to a predefined summarization ratio. We propose to use agenda and speaker-dependent characteristics (such as verboseness, gender, native language, role in the meeting) to improve extractive meeting summarization performance. This is motivated based on the following observations.

- When producing meeting summaries/minutes, humans often start with logistics of the meeting, such as time, place, attendees and presenters. They then use meeting agenda as an outline and build the body of the summary, including important elements such as any decisions made during the meeting. We expect that meeting agendas can play an important role in generating concise meeting summaries by identifying focuses of the meeting and eliminating off-topic discussions.

- Meetings are typically multi-party conversations. Speakers differ in their speaking styles and lexical usage. In addition, a speaker' role in a topic discussion also has an impact on the speaker's speaking style. Different from most text domains where a document is generally written by one person, each participant in the meeting can begin a new topic when starting his/her turn. Hence we expect that a thorough analysis of the speaker characteristics would be beneficial for us to develop speaker-dependent features for summarization.

Section 5.1 describes the experimental setup and the "pseduo-agenda" items we used in the current experiments. In Section 5.2, we introduce the unsupervised Maximum Marginal Relevance (MMR) framework with different vector representations for the meeting: using pseudo-agenda items, the entire meeting transcript, or the combination of the two. In the supervised framework presented in Section 5.3, we perform an in-depth analysis of the discriminative power of features normalized based on speaker information, and then accordingly integrate a variety of speaker-sensitive features in the supervised system. Section 5.4 shows the experimental results. Section 5.5 summarizes this chapter.

5.1 Data Preparation

In a preprocessing step, we first eliminate the DA candidates in the original transcripts that are not likely to be summary DAs. We construct a list of stopwords consisting of 250 and 200 words respectively for human transcripts and ASR output. These words have the lowest IDF values (inverse DA frequency). DAs that only consist of stopwords and functional words are then filtered out. We found on the training set that this filtering process removes 56.36% of the non-summary DA candidates, while only 9.15% of the summary DAs are removed. The ratio between summary and non-summary DAs was reduced from the original 1:14.45 to 1:6.94. The average number of turns in each meeting decreased dramatically, from 833.65 to 264.10. This shows that the prevalent existence of acknowledgment tokens (such as "uh-huh") or backchannels tends to break the conversation discourse into small pieces, hence we believe a filtering process can keep the original speaker turn and will be more useful for our study of speaker turn related features. This preprocessing is applied to both unsupervised and supervised methods. It can have several benefits:

(1) reduce the computational complexity in the MMR approach; (2) remove unlikely summary candidates; (3) improve the balance between the two classes for supervised model training; and (4) better form a speaker turn.

The ICSI meeting corpus does not have the associated meeting agenda. To approximate the agenda, we use the "topic description" in the AMI annotation for the corpus* and refer to them as "pseudo agenda items". These are short phrases generated by human annotators for each topic segment. A sample pseudo-agenda for a meeting is presented in Table 5.1. Note that these topic descriptions were annotated after the meeting, while the real agenda is generally provided before the meeting. Despite this difference, both of them provide a concise outline of the important issues discussed during the meeting, which motivates us to use these pseudo-agenda items in this study. On average, each meeting has about 19.41 pseudo-agenda items (excluding those functional topics such as Digit task, Opening, Closing, Chitchat), and each pseudo-agenda item has 3.55 words.

As we will see in Section 5.2, an exact word match is required for similarity measure. If a pseudo-agenda (or real agenda) word does not appear in the transcript, it will not contribute to the similarity or salience measure for summarization. After removing punctuation, we found that only 81.72% of the agenda word tokens appear in the original meeting transcripts. This rate increases to 85.77% when using the lemmatized transcripts. From the data, we notice that some human written pseudo-agenda words are not in the transcripts because of: different word variations ('recogniser/recognizer', 'bayesnet/bayes-net/bayes net'), spelling errors ('degredation/degradation'), or paraphrase ('create/produce'). To further increase word coverage, we leverage the Google dictionary†. For each uncovered pseudo-agenda word, we create a query to Google dictionary and

*http://corpus.amiproject.org/documentations/annotations

†http://www.google.com/dictionary

Table 5.1. Examples of pseudo-agenda items

Topic ID	pseudo-agenda items
1	neural net test results
2	training data, language, task, feature comparisons
3	noise condition
4	on-line normalization
5	conclusions, decision for next week's testing
6	increasing the number of phonemic classes
7	HTK testing
8	using both features, net outputs
9	OGI voice activity detection (VAD) results
10	derivation of LDA filter
11	baseline ASP
12	application of VAD
13	collaboration with OGI, parameter selection, allocation
14	how many parameters
15	disk resources, servers
16	neural net trainings, HTK runs, processing issues
17	Aurora, large vocabulary training, testing

parse the returned webpage to obtain other variations of this query word. Google dictionary suggests possible correct forms for a mis-spelled word; it also lists verb inflections such as 3rd person present, present participle, past tense, past participle, as well as the plural form for nouns; under the "Web definitions" section, it provides some useful candidates that are semantically related to the query words. If any of the word variations or related words from Google page appears in the original meeting transcript, we use it instead of the original uncovered agenda word. After this expansion process, we achieved a word coverage of 90.05%. Some examples of the originally uncovered words and their covered variations are: polsemy => polysemy, experimental => experiment, selection => choice, bayesnet => bayes, identification => recognition, generalisation => generalization.

5.2 Unsupervised MMR with Pseudo-agenda Information

We choose to use the Maximum Marginal Relevance (MMR) framework due to its simplicity and verified competency in spoken document summarization tasks. We expect this is a good starting point for incorporating the pseudo-agenda information in the meeting summarization system. For each dialogue act S_i, its MMR score $MMR(S_i)$ is the linear combination of its similarity to the original document or a user query ($Sim_1(S_i, D)$) and the similarity to the current selected summary sentences ($Sim_2(S_i, Summ)$), as shown in the following formula:

$$MMR(S_i) = \lambda \times Sim_1(S_i, D) - (1 - \lambda) \times Sim_2(S_i, Summ)$$

Here λ is the balancing factor between the two components. We use cosine similarity under the vector space model for the similarity between two text segments (S_i and S_j):

$$Sim(S_i, S_j) = \frac{\sum_k w_{i,k} \times w_{j,k}}{\sqrt{\sum_k w_{i,k}^2} \times \sqrt{\sum_k w_{j,k}^2}}$$

The term weight for a word $w_{i,k}$ is determined by $\sqrt{TF} \times IDF$, where TF is its term frequency in text segment S_i, and IDF is the inverse document frequency. We use only content words and non-stopwords to form the word vectors.

We refer to the above baseline MMR framework "MMR-Meeting" where we simply use the entire meeting transcripts to form the centroid vector D. We then investigate two other ways to integrate the pseudo-agenda information in the MMR framework: (1) "MMR-Agenda": this one uses the pseudo-agenda words rather than the entire meeting to form the centroid vector; (2) "MMR-Combine": in this method, we use a weighted combined similarity from "MMR-Meeting"

($Sim_1(S_i, D)$) and "MMR-Agenda" ($Sim_1(S_i, A)$) as the first similarity measure, and use a similar MMR formula:

$$MMR_{comb}(S_i) =$$
$$\lambda \times (\mu \times Sim_1(S_i, D) + (1 - \mu) \times Sim_1(S_i, A)) - (1 - \lambda) \times Sim_2(S_i, Summ)$$

where parameter μ is used to balance the weight between the two components, and was tuned on the development set. This approach can also be thought of as using a reweighted vector for D that is formed using the entire meeting transcripts and pseudo-agenda, equivalently giving more weight to the pseudo-agenda words in cosine similarity measure.

5.3 Supervised Framework

We employ a supervised framework to better utilize both pseudo-agenda and speaker-related characteristics. Under this framework, extractive meeting summarization is modeled as a binary classification process. A classifier is trained using the annotated data and assigns posterior probabilities for each DA during testing. The higher ranked DAs are selected in summary. A variety of features have been explored in previous studies. Among them, length, structural, and similarity related cues have been proved to be very effective and form competitive baselines [26, 82, 69]. Therefore we take these length, structural, and similarity information as base features, and expand them to take into consideration different speaker-related attributes and speaking styles in the supervised framework.

5.3.1 Basic Features

- Length and location features (Len + Loc (mt)):

  (A) utterance length, measured by number of words, or seconds (2 features).

  (B) location of the utterance, represented using the portion of utterances before or after the current DA. The portion of utterances can be measured by number of words or DAs, or seconds (6 features).

  We normalize the above base features to $[0, 1]$ by dividing each of them by the maximum obtainable value at the meeting scale: length features are normalized using the longest DA in the meeting; location features are divided by all the utterances in the meeting. We refer to this meeting scale normalization as "mt".

- Similarity features (Sim (mt)):

  (1) "sim-meeting": cosine similarity between a DA and the entire meeting under the vector space model, where the term weight is the same as used in MMR, $\sqrt{TF} \times IDF$. Again only content words and non-stopwords are used to form the vector. (2) "sim-agenda": defined as the cosine similarity between the candidate DA and the pseudo-agenda word vector. We use the same term weighting for the pseudo-agenda vector as in "sim-meeting". (3) "sim-combine": this is the linear combination of the two similarity scores, 'sim-agenda' and 'sim-meeting', with weights determined on the development set[‡]. (4) "sim-combine-smooth": a DA's context can be useful to help determine the importance of the DA, therefore we smooth the combined scores using the following formula:

[‡] $\mu_{agenda} = 0.3, \mu_{meeting} = 0.7$ for human transcript, $\mu_{agenda} = 0.7, \mu_{meeting} = 0.3$ for ASR output

$$sim\_smooth(DA_i) =$$
$$sim(DA_{i-1})/2 + sim(DA_{i+1})/2 + sim(DA_i)$$

where the $sim(\cdot)$ is the "sim-combine" score. We also experimented using more neighboring DAs for smoothing, but did not observe additional gain; therefore only the preceding and the following scores are used in smoothing in this study.

5.3.2 Accounting for Speaker Characteristics

Meetings often have multiple participants. They alternatively present their ideas or thoughts, resulting in an interleaved discourse. We expect a thorough analysis of the speakers' speaking style would help us better understand whether speaker related characteristics have an impact on summarization and how we can effectively utilize speaker information for meeting summarization. Based on data examination, we came up with the following speculative hypotheses that we expect would affect summarization systems.

- Hyp1: Sentence length information should be adjusted based on speakers for summarization. Speakers in a meeting differ in their speaking style (verbose or not) and thus have different tendencies to use long or short DAs. In addition, a speaker's knowledge in a topic discussion also affects that person's style and utterance length.

- Hyp2: Sentence location features should also take into consideration speaker information. Important issues in a meeting might be brought up by different speakers, but for each individual speaker, s/he tends to pose the important DAs at the beginning and conclude at the

end of all of his/her utterances within the entire meeting, or with respect to a specific turn of this speaker.

In order to test our hypotheses and investigate if speaker specific information helps summarization, we introduce two different normalization methods:

- speaker meeting-level normalization (denoted by "spkr"): location features are calculated based on all utterances from the same speaker in the meeting; for each of the length, location, and similarity related features, we divide its feature value by the maximum obtainable feature value among all the utterances by this speaker;
- speaker turn-level normalization (denoted by "turn"): location features are calculated based on utterances from the current speaker turn; each feature value is divided by the maximum available value within this turn. This normalization aims to model the speaker's behavior within a particular speaker turn, thus it is a more local normalization, whereas the first speaker normalization method considers more global behavior from the entire meeting.

Features normalized using these methods will be evaluated in the summarization task in Section 5.4.2. In this section, we evaluate the discriminating power of the above normalized features on the training set using two criteria: (1) "AbsDiff": For each feature, this is the absolute difference between the average value for summary and non-summary DAs. A larger "AbsDiff" value therefore corresponds to better class separability of this feature; (2) Fisher's discriminant ratio C: For a specific feature variable, Fisher's ratio C across two classes i (summary DAs) and j (non-summary DAs) is defined as follows:

$$C = \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2} \tag{5.1}$$

where μ_i, μ_j and σ_i^2, σ_j^2 are means and variances of classes i and j. Larger Fisher's ratio means stronger discriminating power of this feature.

Table 5.2 shows the results using these two metrics for all of the features, normalized based on the meeting, speaker, and speaker turns (denoted as "mt", "spkr", "turn" respectively). For length and location related features, we observed an improved discriminating power after performing speaker level normalization. For location features, using speaker-turn level normalization also increases their ability to separate the summary DAs from the non-summary DAs. The speaker normalized length features promote the DAs from un-verbose speakers; while the normalized location features emphasize on the relative location within the utterances of the same speaker or within the current speaker turn. These results verified our hypotheses. For the similarity features, speaker normalization shows larger "AbsDiff" scores, but generally no gain on the Fisher's ratio (except marginal improvement for "sim-meeting").

The last hypothesis we have is that speaker biographic attributes might affect meeting summarization. For example, DAs could be selected disproportionably from male or female speakers, from native or non-native speakers, from participants with different roles (professors or non-professors in our corpus), with potential favors toward either side. Therefore we performed an analysis to examine the difference of the summary DA percentage corresponding to these factors. Note that different from the above evaluations of speaker-sensitive features, this analysis is not meant to introduce new features, but to examine whether different speaker biographic attributes will affect the chance of an utterance being selected into the summary. We use number of words

Table 5.2. Fisher ratio and AbsDiff of average scores between summary and non-summary DAs for different features, using three different normalization methods.

		AbsDiff			Fisher Ratio		
Normalization		mt	spkr	turn	mt	spkr	turn
Len	second	.082	**.112**	.067	.142	**.153**	.022
	word	.085	**.113**	.068	.131	**.150**	.025
Loc (begin)	second	.054	**.063**	**.061**	.018	**.023**	**.020**
	DA	.052	**.059**	.039	.017	**.021**	.008
	word	.054	**.062**	**.059**	.018	**.023**	**.019**
Loc (end)	second	.054	**.058**	**.071**	.017	**.020**	**.025**
	DA	.052	**.058**	**.085**	.017	**.021**	**.039**
	word	.054	**.058**	**.073**	.018	**.020**	**.027**
Sim	meeting	.080	**.090**	.043	.097	**.099**	.014
	agenda	.079	**.091**	**.136**	.067	.066	.048
	comb	.092	**.104**	.051	.115	.005	.015
	comb-smth	.063	**.072**	.018	.073	.072	.004

and DAs, and seconds to measure the portion of utterances. The resulting scores are presented in Table 5.3. For example, suppose we use the number of DAs as the measurement. Among all the utterances by male speakers, 13.0% of the DAs are included in the meeting summary, while 11.5% of the DAs from female speakers are included in the summary. The difference between these two ratios is .015, which is presented in the table corresponding to the "Male/Female" row and "DA" column. We observe that overall the difference with respect to these factors is rather small. The gender information yields slightly larger difference in summary utterance selection than other attributes. Using number of DAs as the measurement results in larger difference compared to using words or seconds.

Table 5.3. Difference of the summary percentage (measured using second, and number of DAs or words) for different speaker attributes.

Difference		second	DA	word
Biographic Attributes	Male/Female	.012	**.015**	.013
	Native/Non-native	.001	**.005**	.001
	Faculty/Non-faculty	.001	**.002**	.001

5.4 Experiments

5.4.1 Experimental Setup

We use 6 meetings (same as in [26, 36]) from the ICSI corpus as test set; 20 meetings are randomly selected as development set, and the rest 48 meetings[§] are used for training our supervised system. We use the AMI annotations as reference to make our work comparable to the state-of-the-art results. Each test meeting has three human reference summaries, while there is only one reference summary for each of the development and training meetings. We use the TreeTagger[¶] to lemmatize both human transcripts and ASR output, and the TnT part-of-speech tagger [78] trained from Switchboard data for tagging.

We evaluate meeting summarization performance using the following three metrics:

- ROUGE [63]. This has been widely used in prior studies on meeting summarization task. ROUGE scores measure the n-gram overlap between the system summary and a set of human reference summaries. We use ROUGE-1 and ROUGE-2 F-scores to make our results comparable to other previous research.

[§]One meeting (Bed002) was dropped due to poor transcription quality.

[¶]http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

- DA F-score. This one compares the system extracted summary DAs to human annotated ones, and calculates the DA-level precision and recall scores. The DA-level F-measure is then calculated as the harmonic mean of the precision and recall values with equal weights.

- Pyramid approach [83]. Following [36], we use a location-restricted Pyramid score to measure summarization performance. In this approach, a summary content unit (SCU) is defined as a word and its location in the document (index of DA), thus the same word appearing in different locations is discriminated. For example, ("voice", 16) and ("voice", 27) are considered as two different SCUs. The score of each SCU is defined as the total number of times it appears in the human reference summaries. For each system generated summary, we compute a score D by adding up all its SCU scores. A maximum score D^* is calculated as the maximum obtainable SCU scores given the summary length constraint. The Pyramid score is defined as $P = D/D^*$.

5.4.2 Results on Development Set

MMR Approaches

We compare the performance of the three MMR-based approaches on the development set, namely "MMR-meeting", "MMR-agenda", and "MMR-combine". Results on human transcripts are plotted in Figure 5.1.[‖] We compare summaries with word compression ratios ranging from 10% to 30%, catering users with different information need. The best ROUGE-1 and ROUGE-2 F-scores within this range are presented in Table 5.4.

[‖]Results for ASR condition show similar trend as on human transcripts, therefore are not presented.

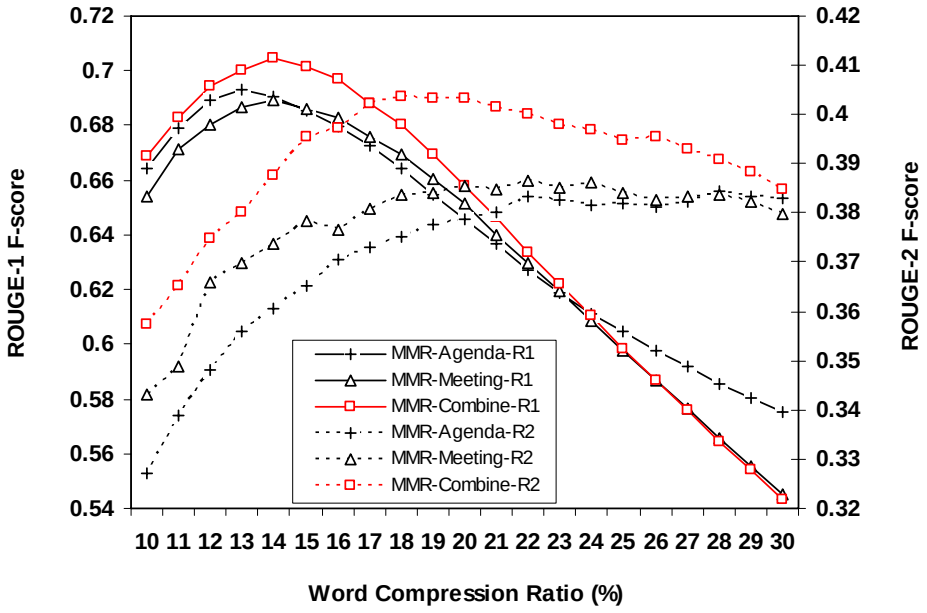


Figure 5.1. MMR results on dev set, with salience score for the DA calculated based on pseudo-agenda items, meeting transcript, or a linear combination of their similarity. Results are for human transcripts, using different compression ratios.

From Table 5.4, we can see that the MMR-combine approach (that combines similarity to both entire meeting and the pseudo-agenda) achieves the best performance on both human transcripts and ASR output. On human transcripts, using MMR-agenda yields even better ROUGE-1 results than MMR-meeting, even though the former has much fewer words. On ASR condition, results using MMR-agenda are inferior to others, suggesting that it is affected more when there are ASR errors. If the agenda words are not correctly transcribed, the corresponding candidate DAs will receive much lower score. We also observed that MMR-agenda approach tends to result in higher precision scores but relatively lower recall scores compared to the MMR-meeting approach. Therefore in future research we may need to expand the current agenda items, such as combining the current pseudo-agenda items with keyphrase annotations.

Table 5.4. MMR results on dev set, with salience score for the DA calculated based on pseudo-agenda items, meeting transcript, or a linear combination of their similarity. Results are for both human transcripts and ASR output, using the best compression ratio.

	Human		ASR	
MMR	R-1	R-2	R-1	R-2
MMR-agenda	69.32	38.43	63.49	28.72
MMR-meeting	68.93	38.67	65.65	29.96
MMR-combine	**70.44**	**40.36**	**66.91**	**31.22**

From Figure 5.1, we can see that overall MMR-combine consistently outperforms others for different compression ratios. Since we limited the DA selection in the MMR-agenda approach to those that have non-zero cosine similarity with the agenda items, we notice that when more DAs are extracted (compression ratio near 30%), the high precision of the MMR-agenda approach leads to its better performance than others. We can also see from the figure that the ROUGE-1 scores for all the approaches peak at a smaller range corresponding to word compression ratio of 15% to 16%, while the highest ROUGE-2 scores scatter on a larger range between the compression ratio of 18% and 25%. These results indicate that, different evaluation metrics may favor different word compression regions. Hence a good summarization approach should either outperform other approaches in all of these ratio ranges to ensure users with different information need are satisfied, or return to the user a summary with optimized length which can achieve the best system performance. We will investigate along these directions in the future studies.

Supervised Approaches

We have demonstrated the effectiveness of speaker based normalization for the features we used in supervised approach in Section 5.3.2. In this experiment, we evaluate speaker normalized

features using the final summarization results. The features are length, location, and similarity related features, as described in Section 5.3. We compare different normalization methods, at the meeting, speaker, or speaker turn level, and also investigate whether combining features with different normalization levels can result in further improvement. All these experiments are performed on the 20-meeting development set. Based on the results in Table 5.2, for length and similarity features, we compare the "mt" and "spkr" level normalization and their combination ("turn" level normalization is not included due to lower discriminating power). For location features, we use the three normalization methods by themselves, and various combination of them. Table 5.5 shows the ROUGE-1 results using 15% compression ratio, with best results for each feature category shown in bold.

For length related features, combining both "mt" and "spkr" level normalization achieves the best performance on human transcripts, while "mt" only normalization results in best performance on ASR output. We then use these two setups as base features for human and ASR transcripts respectively, and add location or similarity features with different normalization levels. For location features, on human transcripts, combining "spkr" and "turn" level normalization works best, and "turn" normalization has the best performance by itself; for ASR output, utilizing all the three levels of normalization or "mt+turn" yields the best results. For similarity features, the best performance is from "spkr" normalization on human transcripts, and "mt" normalization on ASR. The combination of the two normalizations does not yield additional gain. Note that in this experiment our purpose is to examine the effect of different levels of normalization and their combination, rather than performing individual feature selection process. We also experimented with adding binary features indicating whether the DA was from male/female, native/non-native,

or faculty/non-faculty speakers, but did not observe performance gain over the base features.

Table 5.5. Supervised summarization results (ROUGE-1 F-measure) on development set, using features normalized on the meeting, speaker, or speaker turn level, or using various combination of them.

	Normalization	Human	ASR
Len	mt	69.11	**65.56**
	spkr	69.32	65.15
	mt + spkr	**69.47**	65.36
Len + Loc	mt	69.67	65.86
	spkr	69.73	65.77
	turn	70.17	65.78
	mt + spkr	69.52	65.96
	mt + turn	70.33	66.25
	spkr + turn	**70.42**	66.22
	mt + spkr + turn	70.32	**66.25**
Len + Sim	mt	69.73	**66.14**
	spkr	**69.80**	65.95
	mt + spkr	69.72	66.05

In Table 5.6, we summarize the results using supervised methods in order to show the effect of adding pseudo-agenda and speaker normalized features. The "Len + Loc (mt)" approach uses the meeting-level normalized length and location features mentioned in Section 5.3; these features themselves form a very competitive baseline as observed in [27, 69]. The "Len + Loc + Sim (mt)" approach integrates the similarity feature subclasses introduced in Section 5.3, including similarity to the pseudo-agenda and meeting, and the combination of them. The "Len + Loc + Sim (mt,spkr,turn)" approach employed different levels of normalization for length, location, and similarity feature subclasses, selected according to the best results in Table 5.5. Overall, we obtained the best performance when incorporating both pseudo-agenda and speaker information. Note that the absolute improvement over the "Len + Loc (mt)" approach is not very large, but this is competitive comparing to the improvement of ROUGE results reported in other studies for

summarization task. We will show next this gain holds on the test set as well.

Table 5.6. Supervised results (ROUGE-1 and ROUGE-2 F-measure) on development set, w/ or w/o pseudo-agenda and speaker normalized features.

	Human		ASR	
Supervised	R-1	R-2	R-1	R-2
Len + Loc (mt)	69.79	41.58	65.99	32.33
Len + Loc + Sim (mt)	70.00	41.91	66.67	32.61
Len + Loc + Sim (mt,spkr,turn)	**70.68**	**42.22**	**67.17**	**32.65**

5.4.3 Results on Test Set

Finally, we evaluate our MMR and supervised approaches on the test set, using ROUGE-1, ROUGE-2, Pyramid, and the DA-level F-measure as evaluation metrics. ROUGE results are presented in Table 5.7 for different word compression ratios. These ranges correspond to the peak areas using the ROUGE evaluation metric. We use the "MMR-meeting" as a competitive unsupervised baseline. "MMR-comb" is shown to demonstrate the effect of adding pseudo-agenda information in the unsupervised approach. "Supervised-base" uses the meeting-level normalized length and location features, while "Supervised-comb" uses the best combination of all normalization levels as well as the pseudo-agenda related similarity features. We also include an "Oracle" result in this test. For human transcript, this was generated by randomly selecting DAs from the pool of reference summary DAs, until the word compression ratio is reached. This random selection process is repeated 1000 times and an average result is reported. The "Oracle" result for ASR output was from a similar procedure, but using the ASR words with aligned DAs.

For human transcripts, we observed some marginal improvement when adding pseudo-agenda in the MMR approach. Compared to "Supervised-base", better performance was achieved by

Table 5.7. Summarization results (ROUGE-1 and ROUGE-2) on the test set using human transcripts and ASR output.

		R-1 F-score				R-2 F-score						
Word Ratio		14%	15%	16%	17%	19%	20%	21%	22%	23%	24%	25%
Human	Oracle	76.39	76.95	**77.10**	76.89	59.59	60.34	60.98	61.54	62.03	62.45	**62.80**
	MMR-meeting	69.49	69.93	**70.20**	70.05	41.26	41.45	41.75	41.80	41.85	**42.01**	41.71
	MMR-comb	70.39	**70.71**	70.71	70.43	41.26	41.21	41.98	42.02	42.00	**42.14**	42.07
	Supervised-base	70.40	70.42	**70.61**	70.48	43.24	43.55	43.96	**44.04**	44.03	43.92	43.78
	Supervised-comb	71.09	71.69	**71.70**	71.30	45.37	45.31	**45.57**	45.40	45.15	45.07	44.87
ASR	Oracle	70.79	**71.11**	71.07	70.74	42.86	43.28	43.63	43.93	44.18	44.37	**44.53**
	MMR-meeting	65.95	66.49	**66.67**	66.35	31.61	31.65	31.64	**31.67**	31.47	31.59	31.48
	MMR-comb	66.84	**67.08**	67.00	66.68	32.34	32.59	32.71	**32.93**	32.83	32.77	32.61
	Supervised-base	66.75	66.87	**66.98**	66.63	33.32	33.37	33.17	33.32	33.31	**33.46**	33.46
	Supervised-comb	67.70	68.07	**68.09**	67.66	34.47	**34.68**	34.68	34.64	34.50	34.43	34.12

Table 5.8. Results (Pyramid and DA F-score) on the test set using human transcripts and ASR output.

		Pyramid	DA F-Score
Human	MMR-meeting	51.32	31.08
	MMR-comb	51.91	33.16
	Supervised-base	55.65	30.54
	Supervised-comb	**60.32**	**35.33**
ASR	MMR-meeting	40.94	31.38
	MMR-comb	41.23	33.01
	Supervised-base	45.16	32.38
	Supervised-comb	**47.45**	**34.42**

"Supervised-comb" approach that combines pseudo-agenda and speaker information. It reduced the performance gap to the "Oracle" from 6.49 to 5.4 according to ROUGE-1 scores (16.80% relative improvement), and from 18.76 to 17.23 (8.16% relative improvement). We observe similar absolute performance improvement on ASR condition, however, because of the lower "Oracle" results, the reduction of performance gap to "Oracle" is bigger using the "Supervised-comb" approach — 26.88% and 11.02% respectively according to ROUGE-1 and ROUGE-2 measures.

We present the results using the location-restricted Pyramid approach and the DA-level F-measure scores in Table 5.8. These correspond to 19% word compression rate. Again, the combined supervised system achieved the best performance among all the approaches. The performance gain using these two metrics is even larger than that of the ROUGE scores. The consistent improvement of our proposed method across different summarization evaluation metrics suggests the robustness of our approach.

5.5 Summary and Discussions

In this chapter, we investigate using meeting-specific characteristics to improve extractive meeting summarization. Information capturing two aspects is used: pseudo-agenda information, which is generated from topic labels and expected to be effective in eliminating off-topic discussions; and speaker-related attributes (such as verboseness, gender, native language, role in the meeting) were analyzed to help develop a rich set of speaker-sensitive features. We perform experiments on the ICSI meeting corpus using both unsupervised Maximum Marginal Relevance (MMR) framework and supervised approaches. Results are evaluated using multiple criteria, including ROUGE, an approximated Pyramid approach, and a sentence-level F-measure, and show consistent improvement on various testing conditions.

We further performed manual analysis and found that the improved system performance can be partly attributed to the pseudo-agenda and speaker normalized features. Example 1 below shows an example of some off-topic conversations that are effectively eliminated from being selected as summaries based on the pseudo-agenda information. We also noticed that some long DAs pose difficulty for the summarization system. Take the second DA in Example 2 for instance, it just gives some supporting description for the first DA and thus is not considered as summary DA by any of the annotators. However, since this DA is very long and contains some important words, it is selected by the system, even though the speaker turn information gives higher weight to the first DA. In our future work, we will focus on addressing this kind of problems.

Example 1:

[1] spk1: i crashed when i started this morning

[2] spk2: you crashed crashed this morning

[3] spk2: i did not crash this morning

[4] spk1: well maybe it's just you know how many how many times you crash in a day

Example 2:

[1] spk1: it basically says well this is construal

[2] spk1: and then it continues to say that one could potentially build a probabilistic relational model that has some general domain-general rules

CHAPTER 6

FROM EXTRACTIVE TO ABSTRACTIVE MEETING SUMMARIES

Most of the current summarization systems adopt extractive approaches. The system first extracts a set of salient sentences that can best convey the main content of the text document or the spoken audio file; these sentences are then concatenated into a summary according to their appearance in the original document. For the well-formed written text domain, this approach results in quite good summary quality, since the extracted sentences themselves are usually well-formed, self-explainable, and have good sentence and discourse structure. On the other hand, directly concatenating the transcribed utterances may not form a good summary for speech domains. This is especially true for meetings, where disfluencies and redundancies in spontaneous speech significantly affect the readability of the extracted summary.

In Table 6.1, we show an example of the extractive summary and its compressed variant for a meeting dialogue segment. The "Original Extractive Summary" was formed by directly concatenating the extracted summary sentences (using human transcripts). The "Compressed Summary" was generated by manually compressing the extractive summary at the sentence level. We can see that the quality of the original extractive summary is not very good. In contrast, the compressed summary removes many unnecessary words from the original extractive summary. It effectively highlights the main content and its readability is much better. In this sense, the compressed meeting summary is also closer to the abstractive meeting summary. For abstractive summarization, we may apply sentence compression techniques to extracted summary sentences, followed by further

Table 6.1. Human compressed summary sentences for an example meeting dialogue segment. Dialogue act indices (based on the entire meeting) are shown in the first column.

Original Extractive Summary	
423	there there are a variety of ways of doing it
433	so it's possible that we could do something like a summary node of some sort that
444	so what i was gonna say is is maybe a good at this point is to try to informally
446	i mean not necessarily in th- in this meeting but to try to informally think about what the decision variables are
450	and the other trick which is not a technical trick it's kind of a knowledge engineering trick is to make the n- -pau- each node sufficiently narrow that you don't get this combinatorics
Compressed Summary	
423	there are ways of doing it
433	it's possible we could do a summary node
444	good at this point is to try informally
446	to informally think about what the decision variables are
450	make each node sufficiently narrow that you don't get this combinatorics

sentence merging, compaction, and generation.

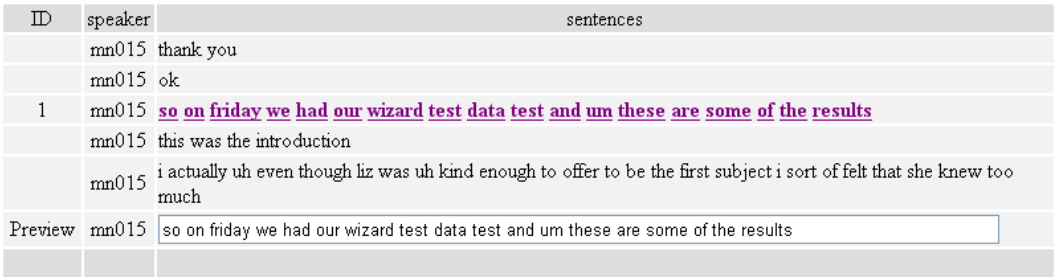
In this chapter, we investigate if we can perform sentence compression on an extractive summary to improve its readability and make it more like an abstractive summary. We propose a fully automatic summarizer that generates compressed meeting summaries by piping the spoken utterance compression module with an extractive meeting summarization system. Two key questions arise in this process. First, is it possible to automatically compress spoken utterances with reasonable performance? To investigate this question, we introduce various compression approaches, including an integer programming (IP) framework with filler phrase (FP) detection module based on Web resources; a noisy-channel approach with Markovization formulation of grammar rules; and

a sequence labeling framework using the conditional random fields (CRF) model to automatically decide whether a word should be kept in the compressed utterance or not. The second question is, should we use the original or the compressed sentences for summary sentence selection? Under the extractive summarization framework, we compare pre-compression and post-compression settings, and evaluate the system performance against the human selected original summary sentences as well as human compressed summaries. Section 6.1 introduces the data corpus; Section 6.2 describes different compression approaches; we pipe the compression module with the summarization system in Section 6.3; Section 6.4 presents the experimental results; finally, we conclude the chapter and present discussions in Section 6.5.

6.1 Data Annotation

We used 26 meetings (same as used in the keyword extraction experiments [84]) for spoken utterance compression annotations. 6 of the meetings are the commonly used test set for meeting summarization systems ([26, 29, 36]), which contain 1088 extractive summary sentences from three annotators. The rest of the 20 meetings have only one summary annotation with 1773 extractive summary sentences. In a preliminary study, we used an annotation set ("Compression Set I") consisting of 455 extractive summary sentences from 6 test meetings labeled by one annotator; later a larger data set ("Compression Set II") was constructed using all the human annotated summary sentences from the 26 meetings (2861 summary sentences) for large-scale utterance compression annotation. Since "Set I" is a subset of "Set II", we use the latter to demonstrate the annotation procedure.

The data annotation was conducted via the Amazon Mechanical Turk (AMT)*. The summary sentences are grouped into 286 human intelligence tasks (HITs); each HIT contains 10 sentences that need to be compressed. Filled pauses such as "uh/um/eh" are removed in the preprocessing step to increase the sentence readability for human annotators. The compression guideline we used is similar to [85]. The annotators were asked to only remove words from the original sentence while preserving most of the important meanings, and make the compressed sentence as grammatical as possible. The annotators can leave the sentence uncompressed if they think no words need to be deleted; however, they were not allowed to delete the entire sentence. Since the meeting transcripts are not as readable as other text genres, we may need a better compression guideline for human annotators. Currently we let the annotators make their own judgment what is an appropriate compression for a spoken sentence.



ID	speaker	sentences
	mn015	thank you
	mn015	ok
1	mn015	so on friday we had our wizard test data test and um these are some of the results
	mn015	this was the introduction
	mn015	i actually uh even though liz was uh kind enough to offer to be the first subject i sort of felt that she knew too much
Preview	mn015	so on friday we had our wizard test data test and um these are some of the results

Figure 6.1. Annotation Interface

We use a two-stage annotation scheme. In the first stage, each HIT was annotated by 8 mechanical turk workers. Each received $0.15 as compensation for every HIT. The annotation interface we used is shown in Figure 6.1. For each sentence that needs to be compressed, two sentences before and after it are displayed in the annotation interface in order to provide some context. We also

*http://mturk.com

show the speaker id for all the sentences since this is a multi-party conversation and knowing who said it is helpful to understand the utterance. The turkers can click on the unnecessary words and remove them from the original sentence; the resulting compressed sentence is shown in a preview text box. In total, 244 turkers participated in the first stage annotation. The average working time for compressing 10 sentences (one HIT) is 4.29 minutes.

In this study, for each utterance we only use one compression, the best compression found from the 8 annotations from the first stage. We use AMT again to conduct a second-stage annotation to find the best compression: we provide the same original summary sentence and its context to the annotators as in the first stage, list all the compression variants, and ask the turkers to select the best compression for each original summary sentence. Each sentence is annotated by 6 turkers in this annotation stage. Their majority vote is used as the gold standard compression. If there is a tie, we choose the shorter one. It takes 4 minutes on average for a turker to select the best compressions for 10 sentences. 300 turkers performed the second stage annotation. Only 41 of them are the same as in the first stage.[†] We found from the data that 16.12% of the selected best compressions are agreed on by all of the 6 annotators; 21.07% are agreed by 5 annotators; 28.70% are agreed by 4 annotators; 27.99% are agreed by 3 annotators; 6.09% are agreed by 2 annotators, and 0.03% by 1 annotator.

[†]We did not explicitly require different sets of turkers for these two stages. Given the large number of turkers in the two annotation stages and the number of HITs in our task, the chances that a turker works on the same set of utterances in the two stages (thus may be biased in the selection of best compressions in the second stage) are quite small.

6.2 Spoken Utterance Compression Approaches

Previous studies of sentence compression mostly focus on compressing the well-formed sentences from the news documents. [86, 87] learned rewriting rules that indicate which words should be dropped in a given context. [86, 88] applied the noisy-channel framework to predict the possibilities of translating a sentence to a shorter word sequence. [89] extended the noisy-channel approach and proposed a head-driven Markovization formulation of synchronous context-free grammar (SCFG) deletion rules. Unlike these approaches that need a training corpus, [85] encoded the language model and a variety of linguistic constraints as linear inequalities, and employed the integer programming approach to find a subset of words that maximize an objective function. [90, 91, 92] proposed a generic sentence trimmer that consists of a generation component and a reranking component. The goal is to first generate a set of grammatically correct compression candidates, then select the best compression among them using a conditional random fields model. The training data are collected from the online RSS feeds, therefore avoiding the human efforts for generating gold standard compressions. As to the spoken language domain, [93] proposed to compress dialogue acts by preserving the pitch contour of the original whole utterance, and achieved promising results on a corpus of 30 dialogue acts from the ICSI meeting corpus.

In this section, we introduce several spoken utterance compression models, including an integer linear programming framework with filler phrase detection, a noisy-channel approach with Markovization formulation of grammar rules (as in [89]), and a CRF model that incorporated word identity features, part-of-speech (POS) features, position features, and features derived from both syntactic and discourse parsing trees.

6.2.1 Compression Using Integer Programming

We employ the integer programming (IP) approach in the same way as [85]. Given an utterance $S = w_1, w_2, ..., w_n$, the IP approach forms a compression of this utterance only by dropping words and preserving the word sequence that maximizes an objective function, defined as the sum of the significance scores of the consisting words and n-gram probabilities from a language model:

$$\max \ \lambda \cdot \sum_{i=1}^{n} y_i \cdot Sig(w_i) + \ (1-\lambda) \cdot \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} x_{ijk} \cdot P(w_k|w_i, w_j)$$

where y_i and x_{ijk} are two binary variables: $y_i = 1$ represents that word w_i is in the compressed sentence; $x_{ijk} = 1$ represents that the sequence w_i, w_j , w_k is in the compressed sentence. A trade-off parameter λ is used to balance the contribution from the significance scores for individual words and the language model scores. More details can be found in [85]. We only used linear constraints defined on the variables, without any linguistic constraints.

We use the lp_solve toolkit[‡]. The significance score for each word is its TF-IDF value (term frequency $\times$ inverse document frequency). We train a language model using SRILM [§] on broadcast news data to generate the trigram probabilities. λ is empirically set as 0.7, which gives more weight to the word significance scores.

As an important first step, we propose a filler phrase detection module, and filter them out from the sentences before applying the integer programming approach. We define filler phrases (FPs) as the combination of two or more words, which could be discourse markers (e.g., I mean, you know), editing terms, as well as some terms that are commonly used by human but without critical

[‡]http://www.geocities.com/lpsolve

[§]http://www.speech.sri.com/projects/srilm/

meaning, such as, “for example”, “of course”, and “sort of”. Removing these fillers barely causes any information loss. We propose to use web information to automatically generate a list of filler phrases and filter them out in compression.

For each extracted summary sentence, we use it as a query to Google and examine the top N returned snippets (N is 400 in our experiments). The snippets may not contain all the words in a sentence query, but often contain frequently occurring phrases. For example, “of course” can be found with high frequency in the snippets. We collect all the phrases that appear in both the extracted summary sentences and the snippets with a frequency higher than three. Then we calculate the inverse sentence frequency (ISF) for these phrases using the entire ICSI meeting corpus. The ISF score of a phrase i is:

$$isf_i = \frac{N}{N_i}$$

where N is the total number of sentences and N_i is the number of sentences containing this phrase. Phrases with low ISF scores mean that they appear in many occasions and are not domain- or topic-indicative. These are the filler phrases we want to remove to compress a sentence. The three phrases we found with the lowest ISF scores are “you know“, “i mean” and “i think”, consistent with our intuition.

We also noticed that not all the phrases with low ISF scores can be taken as FPs (“we are” would be a counter example). We therefore gave the ranked list of FPs (based on ISF values) to a human subject to select the proper ones. The human annotator crossed out the phrases that may not be removable for sentence compression, and also generated simple rules to shorten some phrases (such as turning “a little bit” into “a bit”). This resulted in 50 final FPs and about a

Table 6.2. Partial list of collected filler phrases.

you know	and i think	some of
i mean	so far	it seems like
i think	more or less	of course
sort of	or whatever	and so on
kind of	at all	so forth
or something	for example	the fact that

hundred simplification rules. Examples of the final FPs are listed in Table 6.2. When using this list of FPs and rules for sentence compression, we also require that an FP candidate in the sentence is considered as a phrase in the returned snippets by the search engine, and its frequency in the snippets is higher than a pre-defined threshold.

The IP compression method is applied to the sentences after filler phrases (FPs) are filtered out. We refer to the output from this approach as "FP + IP".

6.2.2 Compression Using Lexicalized Markov Grammars

Another sentence compression method we use is the lexicalized Markov grammar-based approach [89] with edit word detection [94]. Two outputs were generated using this method with different compression rates (defined as the number of words preserved in the compression divided by the total number of words in the original sentence).[¶] We name them "Markov (S1)" and "Markov (S2)" respectively.

[¶]Thanks to Michel Galley for generating these output.

6.2.3 Compression Under Sequence Labeling Framework

In this approach, we follow the sentence compression approach in [90] and formulate the spoken utterance compression task as a sequence labeling problem. We label a word with "0" if it is to be removed from the original utterance, and "1" if it is retained.

Given an original word sequence $X = (X_1, X_2, ..., X_n)$, the distribution of its corresponding label sequence $Y = (Y_1, Y_2, ..., Y_n)$ under the linear chain CRF model takes the following form:

$$p(Y|X) \propto \exp \sum_{k=1}^{n} (\sum_{j} \lambda_j f_j(y_k, y_{k-1}, X) + \sum_{i} \mu_i g_i(x_k, y_k, X))$$

where f_j are transition feature functions; g_i are observation feature functions; λ_j and μ_i are their corresponding weights.

We define the features g_i using a set of word tokens, part-of-speech tags, position features, and features extracted from the syntactic and discourse parsing trees.

- Word tokens

  This set of feature templates includes the identity of the current word token; the two word tokens before and after the current word; and all the bigrams and trigrams that can be formed by adjacent tokens and the current word.

- Part-of-Speech (POS) tags

  This set of feature templates includes the POS tags and the tag combinations that correspond to the unigram, bigram, and trigram word token features. We use the TnT part-of-speech tagger [78] trained from Switchboard data for tagging.

- Utterance length features

  There are two features: the length of the current utterance (measured by the total number of word tokens in it), and the relative position of the current word within the utterance (defined as the word position divided by the utterance length).

- Syntactic parsing tree based features

  We use the Charniak's reranking parser[‖] to generate the sentence-level syntactic parsing tree for each utterance. We derive three types of feature templates from the syntactic parsing tree: (1) the second-to-last syntactic tag along the path from the root to the word, which denotes whether the current word token is included in the NP, VP, PP, ADVP phrases. We also include the same context tag information as defined for word token and POS feature templates. (2) length of the path starting from "S1" to the current word. (3) length of the path divided by the longest available path in the current parsing tree.

- Discourse parsing tree based features

  We use the sentence-level discourse parser "SPADE"[**] to generate the discourse parsing tree, whose leaves correspond to elementary discourse units and internal nodes correspond to discourse spans. We generate three types of feature templates from the discourse parsing tree: (1) length of the discourse unit containing the current word, measured by the number of word tokens in it. (2) relative position of the current word token within its discourse segment. (3) the first discourse tag ("Satellite" or "Nucleus") along the parsing tree.

We avoided deriving more complex features due to the concern that the POS tagger, syntactic

[‖] http://www.cs.brown.edu/~ec/#software

[**] http://www.isi.edu/licensed-sw/spade/

and discourse parsers do not perform very well on the ill-formed spoken utterances.

6.3 Using Compression for Meeting Summarization

We choose to use the maximum marginal relevance (MMR) framework due to its simplicity and verified competency in speech summarization. There are a variety of ways to combine the compression with the summarization modules. In this study, we investigate using the compressed sentences (pre-compression) vs. the original transcripts (post-compression) for summarization.

- **Pre-compression**: we perform utterance compression on the original transcripts, then apply the MMR based summarization system on the compressed transcripts. For the summary output, we can use the selected compressed sentences, or map these sentences back to their corresponding original transcripts.

- **Post-compression**: we apply the MMR based summarization system on the original meeting transcripts. In this approach, to generate a compressed summary, there are two methods: (i) we can compress the selected summary sentences (these are in their original uncompressed format); (ii) we can simply map the selected summary sentences to their compressed version if sentences have been pre-compressed already.

We evaluate different configurations (in terms of MMR input and summary output) in order to answer the question whether using compressed sentences helps select better summary sentences for different summarization goals.

6.4 Experiments

In this section, we present the experimental results of the compression approaches, evaluated by human judges, ROUGE metrics, and the compression accuracy. We also evaluate the impact of utterance compression on the meeting summarization task using both pre-compression and post-compression configurations.

6.4.1 Compression Results

A. Results on Set I

First we perform human evaluation for the compressed sentences. Again we use the Amazon Mechanical Turk for the subjective evaluation process. For each extractive summary sentence, we asked 10 human subjects to rate the compressed sentences from the three systems, as well as the human compression. This evaluation was conducted on three meetings of the "Compression Set I", containing 244 sentences in total. Participants were asked to read the original sentence and assign scores to each of the compressed sentences for its informativeness and grammaticality respectively using a 1 to 5 scale. An overall score is calculated as the average of the informativeness and grammaticality scores. Results are shown in Table 6.3. For a comparison, we also include the ROUGE-1 F-scores [63] of each system output against the human compressed sentences.

We can see from the table that as expected, the human compression yields the best performance on both informativeness and grammaticality. 'FP + IP' and 'Markov (S1)' approaches also achieve satisfying performance under both evaluation metrics. The relatively low scores for 'Markov (S2)' output are partly due to its low compression rate (see Table 6.5 for the length information). Exam-

Table 6.3. Human evaluation results. Also shown is the ROUGE-1 (unigram match) F-score of different systems compared to human compression.

Approach	Info.	Gram.	Overall	R-1 F (%)
Human	4.35	4.38	4.37	-
Markov (S1)	3.64	3.79	3.72	88.76
Markov (S2)	2.89	2.76	2.83	62.99
FP + IP	3.70	3.95	3.82	85.83

ples of the compressed sentences can be found in Table 6.4.

Since our goal is to answer the question if we can use sentence compression to generate abstractive summaries, we compare the compressed summaries, as well as the original extractive summaries, against the reference abstractive summaries. The ROUGE-1 results along with the word compression ratio for each compression approach are shown in Table 6.5. We can see that all of the compression algorithms yield better ROUGE score than the original extractive summaries. Take Markov (S2) as an example. The recall rate dropped only 8% (from the original 66% to 58%) when only 53% words in the extractive summaries are preserved. This demonstrates that it is possible for the current sentence compression systems to greatly condense the extractive summaries while preserving the desirable information, and thus yield summaries that are more like abstractive summaries. However, since the abstractive summaries are much shorter than the extractive summaries (even after compression), it is not surprising to see the low precision results as shown in Table 6.5. We also observe some different patterns between the ROUGE scores and the human evaluation results in Table 6.3. For example, Markov (S2) has the highest ROUGE result, but worse human evaluation score than other methods.

To evaluate the length impact and to further make the extractive summaries more like abstractive summaries, we conduct an oracle experiment: we compute the ROUGE score for each of the

Table 6.4. Compression examples.

Approach	Compression
Original	and language input for example is of course crucial you know also when you do the sort of deep understanding analysis that we envision
Human	language input is crucial when you do deep understanding analysis
Markov (S1)	language input for example is of course crucial
Markov (S2)	language input for example is crucial
FP + IP	language input is crucial also when you do the deep understanding analysis that we envision
Original	um we have to refine the tasks more and more which of course we haven't done at all so far in order to avoid this rephrasing
Human	we have to refine the tasks in order to avoid rephrasing
Markov (S1)	we have to refine the tasks more and more which we haven't done in order to avoid this rephrasing
Markov (S2)	we have to refine the tasks which we haven't done order to avoid this rephrasing
FP + IP	we have to refine the tasks more and more which we haven't done to avoid this rephrasing
Original	and uh my suggestion is of course we we keep the wizard because i think she did a wonderful job
Human	my suggestion is we keep the wizard because she did a wonderful job
Markov (S1)	my suggestion is we the wizard because she did a wonderful job
Markov (S2)	we the wizard she did a wonderful job
FP + IP	my suggestion is we keep the wizard because she did a wonderful job

Table 6.5. Compression ratio of different systems and ROUGE-1 scores compared to human abstractive summaries.

	All Sent.				Top Sent.		
Approach	Ratio (%)	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
Extractive summary	100	7.58	66.06	12.99	29.98	34.29	31.83
Human compression	65.58	10.43	63.00	16.95	34.35	37.39	35.79
Markov (S1)	67.67	10.15	61.98	16.41	34.24	36.88	35.46
Markov (S2)	53.28	11.90	58.14	18.37	32.23	34.96	33.49
FP + IP	76.38	9.11	59.85	14.78	31.82	35.62	33.57

extractive summary sentences (the original sentence or the compressed sentence) against the abstract, and select the sentences with the highest scores until the number of selected words is about the same as that in the abstract.[††] The ROUGE results using these selected top sentences are shown in the right part of Table 6.5. There is some difference using all the sentences vs. the top sentences regarding the ranking of different compression algorithms (comparing the two blocks in Table 6.5).

From Table 6.5, we notice significant performance improvement when using the selected sentences to form a summary. These results indicate that, it may be possible to convert extractive summaries to abstractive summaries. On the other hand, this is an oracle result since we compare the extractive summaries to the abstract for sentence selection. In the real scenario, we will need other methods to rank sentences. Moreover, the current ROUGE score is not very high. This suggests that there is a limit using extractive summarization and sentence compression to form abstractive summaries, and that sophisticated language generation is still needed.

[††]Thanks to Shasha Xie for generating these results.

B. Results on Set II

For the CRF-based spoken utterance compression model, we use the CRF++[‡‡] implementation and perform experiments using the "Compression Set II". Training data are the human compressed summary sentences in the 20 training meetings, which contain 1,772 sentences (26,002 word tokens). The test data is from the 6-meeting test set, consisting of 1,088 sentences (16,361 word tokens). In order to generate output with different compression ratios, we use the model's posterior probabilities. For every token, we calculate its confidence score: $p(keep) - p(delete)$, where $p(keep)$ and $p(delete)$ are the posterior probabilities of keeping and deleting this token respectively. We rank all the tokens on the test set according to this confidence measure and preserve tokens with high confidence scores until reaching the specified compression ratio. Note that we choose to use this corpus level compression ratio rather than at the sentence level, since different sentences may need different degrees of compression.

The utterance compression performance is evaluated using the token-level labeling accuracy and f-measure score, as well as the sentence-level labeling accuracy. The token-level accuracy is defined as the number of correctly labeled tokens divided by the total number of tokens in the test set. The f-score is the harmonic mean of precision and recall scores, using preserved tokens as the positive target class. The sentence-level labeling accuracy is the number of correctly compressed sentences divided by the total number of sentences in the test set. This is a more strict measure than the token level ones. Table 6.6 shows the utterance compression results using different compression ratios.

We can see that a high compression ratio corresponds to high recall score and low precision,

[‡‡]http://crfpp.sourceforge.net/

Table 6.6. Spoken utterance compression results on the test set using CRF models with different compression ratios.

	token level				sent level
Ratio	P(%)	R(%)	F(%)	Acc(%)	Acc(%)
0.5	82.60	66.82	73.88	70.79	12.68
0.6	80.09	77.72	78.89	74.29	16.64
0.7	**77.21**	**87.41**	**81.99**	**76.27**	**19.85**
0.8	73.03	94.50	82.39	75.03	17.83
0.9	67.86	98.78	80.45	70.33	10.94

which is expected. When we retain 70% of the total words, the compression system achieved the best performance in terms of both accuracy and f-score, as well as the sentence level accuracy. This is also the compression ratio that is closest to that of the default CRF output, that is, the model determines whether a token is preserved or not without any given compression ratio. The compression ratio based on this default classifier output is 69.76%.

Regarding the features used in the CRF for sentence compression, we notice that the word identity and POS features are strong indicators of unnecessary words, and adding position related features and features extracted from syntactic and discourse parsing tree yielded slight performance improvement. Similar findings are also reported in [90].

6.4.2 Summarization Results

A crucial question we raise is, does compressing all the utterances in the transcripts before performing extractive meeting summarization help summary sentence selection? what is the best system setup for generating compressed extractive summaries? To answer these questions, we pre-compress the utterances in the original transcripts using the above CRF models with different word compression ratios. Then we apply the MMR approach for utterance selection using either

pre-compressed transcripts or the original transcripts as input.[§§] Summarization performance is evaluated using the widely adopted ROUGE score measurement [63].

In the first experiment, we evaluate the utterance selection performance of using both pre-compressed transcripts and original transcripts. When using the pre-compressed transcripts as MMR input, we map the selected summary sentences to their corresponding sentences in the original transcripts. The generated summaries are therefore compared against the human annotated extractive summaries. The summary length is set to be 15% of the total words in the original transcripts. This summarization ratio is similar to those used in previous work for meeting summarization. Results are shown in Table 6.7. Both ROUGE-1 and ROUGE-2 f-scores are presented to make our results comparable with previous studies. Higher ROUGE scores represent better utterance selection performance. We notice that when using compressed sentences in MMR with relatively high compression ratios (deleting limited words), there is moderate improvement, especially when measured by ROUGE-2 scores. We also evaluated using other summary length (shorter and longer), and found that in general, the difference in ROUGE-1 scores is rather small using the two different MMR inputs, and that ROUGE-2 results are slightly better using the compressed sentences in MMR.

In the second experiment, we evaluate the performance of both pre-compression and post-compression in generating compressed meeting summaries. For pre-compression, we use the pre-compressed transcripts as input for the MMR system, then the top-ranked sentences are directly concatenated to form the compressed summaries until a pre-defined summary length is reached. For post-compression, we use the original transcripts as input for the MMR system, then map

[§§] We used fixed parameter $\lambda = 0.5$ for all MMR experiments

Table 6.7. Summarization results using both pre-compressed and original meeting transcripts. In both cases, selected sentences are mapped to the original sentences and compared against human annotated extractive summaries.

	Ratio	R-1 F(%)	R-2 F(%)
MMR: Pre-compressed trans	0.5	70.19	36.00
	0.6	70.60	35.94
	0.7	71.07	**37.06**
	0.8	**71.21**	**37.75**
	0.9	**71.20**	**36.87**
MMR: Original trans		71.12	36.21

the top-ranked summary sentences to their corresponding compressed version to create the final compressed summary.

For this experiment, we use 10% summarization ratio (a smaller number than the previous experiment since the output is a compressed summary). All the generated summaries are compared against human compressed meeting summaries. ROUGE results are shown in Table 6.8. For a comparison, we also include results just using the original transcripts for summarization (with the same summary length, 10%), without any compression. We can see that both pre-compression and post-compression perform much better than simply using the original un-compressed extractive summaries (last row in Table 6.8). Between pre-compression and post-compression approaches, their difference is not very significant. The best results are achieved when using the pre-compression configuration, with different compression ratios for ROUGE-1 and ROUGE-2 measures. There is a larger improvement based on ROUGE-2 results than ROUGE-1 (bold numbers in the table).

As mentioned in Section 6.3, another alternative to perform post-compression is to compress the selected summary sentences, rather than mapping them to pre-compressed sentences. Take 70% utterance compression ratio as an example, we first selected important sentences containing

Table 6.8. Summarization results using pre-compression, post-compression, and no-compression (extractive summary sentences are rendered using original transcripts). The generated summaries are compared against human compressed meeting summaries.

	Pre-compression MMR: compressed sents		Post-compression MMR: original	
Ratio	R-1 F(%)	R-2 F(%)	R-1 F(%)	R-2 F(%)
0.5	62.78	25.06	63.55	25.07
0.6	64.29	**27.14**	64.83	26.37
0.7	**65.30**	**27.77**	65.01	27.27
0.8	64.95	**28.26**	65.09	27.50
0.9	**64.26**	**26.95**	64.03	26.13
Orig trans	R-1 F(%): 60.38 R-2 F(%): 22.86			

14.3% of the total words using the original transcripts as input, and then applied 70% compression ratio to these sentences, thus resulting in 14.3% $\times$ 70% $\approx$ 10% final summarization ratio. We found the results from this summarization-compression pipeline are slightly worse than those in Table 6.8, e.g., yielding 64.97 and 27.58 for R-1 and R-2 respectively for 70% compression ratio. The difference between the two ways of post-compression implementation lies in the definition of the utterance compression ratio, whether it is corpus based or at the sentence level. Further analysis is still needed to understand what is the best set up to generate compressed summaries.

Our experimental results indicate that overall using compressed sentences for summarization performs similarly or slightly better for both summarization goals: generating original extractive summaries and compressed ones. It is also worth mentioning that when summary length is pre-specified (in terms of number of words), using compressed summaries allows the system to include more sentences in the summary, increasing the coverage of important content. This is especially true for conversational speech, in which many words can be removed without significantly affecting information content. The generated compressed summaries improve readability and are closer

to abstractive summaries.

We have performed experiments using different compression ratios above. When compression ratios are high, only a small percentage of words are removed. We expect that those words are likely to be disfluencies. The human annotators typically first remove repetitions, revisions, and other disfluencies to simplify the sentence, then remove other words/phrases to further compress it. In this section, we compare sentence compression with disfluency removal for summarization. [95] used human annotated disfluencies and evaluated the effect of disfluency removal on meeting summarization. They showed that removing disfluencies first, followed by MMR extractive summarization, did not help summary sentence selection. Our results (as shown in Table 6.7 when compared to human extractive summaries) are similar, especially when measured using ROUGE-1 F-scores, but we observed some improvement using compressed sentences for summarization based on ROUGE-2 results.

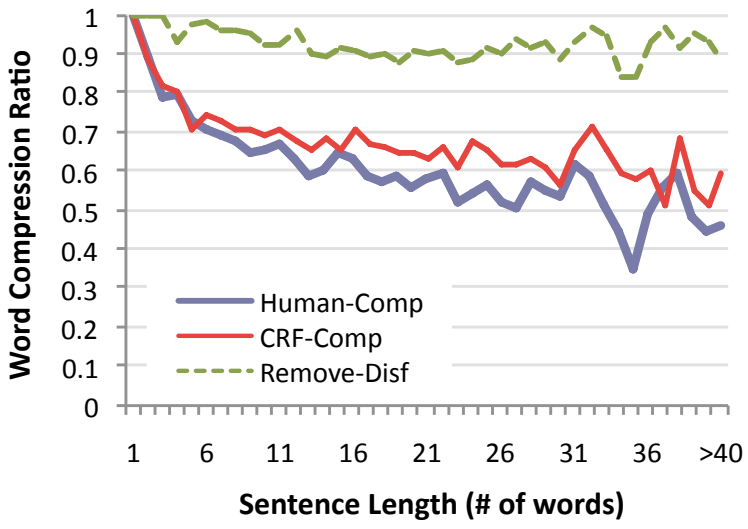


Figure 6.2. Average sentence level word compression ratio with respect to different sentence length.

Liu et al. [95] used the same 6-meeting test set, therefore we can compare their disfluency annotation with our utterance compression annotations on the same summary sentences of these 6 meetings. The corpus-level word compression ratio is 91.58% for disfluency cleaned-up data, and 58.12% for our compression data.[¶¶] In Figure 6.2 we show the average sentence level word compression ratio with respect to different sentence length. We notice that the word compression ratios for disfluency cleaned-up data do not change much for sentences with different length; while for the gold standard compression results, there is a clear trend that longer utterances tend to be compressed more. We also include the analysis for the automatic compression output using the CRF model (with 70% compression ratio) in the figure for a comparison. The compression model shows similar tendency as human compressions – compressing longer utterances more aggressively. This relationship between sentence length and compression ratio also explains why we use corpus level compression ratio instead of sentence level for automatic utterance compression. The above analysis was conducted using the human disfluency annotation results. In the future, we will compare the automatic generated output of a disfluency removal module and an utterance compression module.

6.5 Summary and Discussions

In this chapter, we proposed to automatically generate compressed meeting summaries and improve summarization quality. We investigated the utterance compression task using various approaches, including the integer programming (IP) framework, where we also introduce a filler

[¶¶]Note that these word compression scores are calculated with respect to the original transcripts, while our word compression ratios used for the CRF models were calculated based on the pre-cleaned data (removing "uh/um/eh" for annotation purpose). There is about 3.5% difference between these two measures.

phrase (FP) detection module based on the Web resources; the lexicalized Markov grammar-based approach that considers the grammaticality of the compressed sentences; as well as modeling utterance compression as a sequence labeling problem. We showed satisfying performance using the three approaches, especially the supervised CRF model that incorporates word identity, part-of-speech, position, and features extracted from syntactic and discourse parsing tree. The CRF-based utterance compression module was combined with an MMR based extractive meeting summarization system. We compared using different sentence inputs in MMR: original vs. compressed sentences, and found that the latter performs slightly better, but the difference between the two setups is rather small. Overall, we demonstrated that spoken utterance compression is feasible and that we can generate compressed summaries with reasonable performance. This is one step towards automatic abstractive summarization. In addition, another important contribution of this work is the corpus of compressed spoken utterances we created, which can be used for cross-genre studies.

The compression approach we used in this study can be improved in many ways. For example, we did not explicitly consider the sentence structure. In the future, we may first generate a set of syntactically well-formed utterances, then select the best compression from them. We can also experiment with incorporating prosodic features in the CRF models. Other than the 1-best gold standard compression we used in the experiments, we may consider other alternative compressions. Finally, the automatically compressed utterances will also be manually evaluated for grammaticality and informativeness, as conducted in [96].

Another line of work for summarization is to investigate the possibilities of jointly optimizing the compression and summarization systems. It is possible to let the summarization system decide

whether to use the compressed or the original sentences [97], or whether a particular word should be removed or not in order to generate a high-quality summary. In the future, we also plan to investigate cross-sentence fusion to generate more coherent abstract-alike summaries.

CHAPTER 7

EXPLORING NEW TERRITORY: TWITTER TOPIC SUMMARIZATION

User contributed content has become a major source of information in the Web 2.0 era. People follow their topics of interest, share their experience or opinions on a variety of interactive platforms, including forums, blogs, microblogs, social networking sites, etc. To keep track of the trends online and suggest topic of interest to the general public, many leading websites provide a "buzzing" service by publishing the current most popular topics on their entrance page and update the column regularly, such as the "popular now" column on Bing.com, "trending topics" on Twitter.com, "trending now" on Yahoo.com, Google Trends, and so forth. Often popular topics are in the form of a list of keywords or phrases*; clicking on each phrase will trigger a search request with a set of Twitter posts (tweets) or web pages returned as a result. Nonetheless, whether this is a convenient way for users to navigate through the popular topic information is still arguable. For example, when "SXSW" was listed as a trending topic, it seems difficult to understand at the first glance. A condensed topic summary would be extremely helpful for the users before diving into the massive search results to figure out what this topic phrase is about and why it is trending.

In this chapter, our goal is to generate a short text summary for any given Twitter topic phrase. Different from traditional written text summarization that takes single input source, we propose to utilizing multiple available information sources: the user contributed tweets, normalized tweets via a dedicated twitter message normalization system, web contents linked from the URLs embedded

*They are referred to as topic phrases hereafter, with no distinction between keywords and key phrases.

in the tweets, as well as the combination of different resources. In Table 7.1, we show example tweets and one clip of the linked web content for the topic "SXSW". The tweets were extracted by searching the Twitter site using the topic phrase as query. As can be seen, these sources exhibit vastly different text quality, which poses great challenges to the summarization task.

Table 7.1. Example tweets and a clip of the linked web content for Twitter topic "SXSW".

Twitter Topic: "SXSW"	
Twts	I wish I could go to SXSW... I will, one day! *http://sxsw.com/*
	RT @user123: SXSW Film Round-Up: Documentaries *http://bit.ly/fg033b*
	@user456 yo.whats good,i met u at sxsw, talkin bout that feature.I was gonna see about sending u a few beats.u lookin for only original?
Web Cont	The South by Southwest (SXSW) Conferences & Festivals offer the unique convergence of original music, independent films, and emerging technologies...*(http://sxsw.com/)*

We focus on two questions that are not studied in previous literature: (1) will the web content linked from the tweets be utilized for summarization? Can we integrate different text sources, such as the tweets and linked web pages, to generate more informative Twitter topic summaries? (2) what is the effect of noisy nonstandard tokens on the summarization performance? Will the summaries be improved if the noisy tweets were pre-normalized into standard English sentences? We propose a novel letter transformation approach to convert the nonstandard tokens in the tweets into standard English words. We also investigate the summarization performance by constructing a concept-based summarization framework, utilizing text input with various quality and originated from multiple sources, as well as thoroughly analyzing the resulting summaries using both au-

tomatic and human evaluation metrics. Next, we present the data collection process in Section 7.1; the proposed twitter message normalization system is introduced in Section 7.2; Section 7.3 describes the concept-based summarization system with various text inputs; results and analysis for the normalization and summarization systems are presented in Section 7.4; we conclude the chapter in Section 7.5.

7.1 Data Collection

We collected 5,537 topic phrases and the reference topic descriptions by crawling the Twitter.com and WhatTheTrend.com simultaneously during the period of Aug 22th, 2010 to Oct 30th, 2010 (about 70 days). The Twitter API was queried every 5 minutes for the current top ten trending topics. For each of these topics, a search query was submitted to the Twitter Search API to retrieve only English tweets related to this topic. If any tweet contains embedded URLs linked to the other web pages, the contents of these web pages were retrieved. We limit the maximum number of retrieved tweets and web pages for each topic to 5,000 and 100 respectively due to the space limit. WhatTheTrend API provides short topic descriptions contributed and constantly updated by the Twitter users. There is also a manually assigned category tag for each topic phrase. We found the top categories among the collected topics are "Entertainment (29.26%)", "Sports (25.58%)", and "Meme (15.69%, pointless babble)". We divide the collected topics into two groups: the general topics (e.g., "Chilean miners", "MTV VMA") and the hashtag topics that start with the "#" (e.g., "#top10rappers", "#octoberwish").

To generate reference summaries for the Twitter topics, two human annotators were asked to pick the topic descriptions/sentences (collected from WhatTheTrend.com) that are appropriate and

valuable to be included in the summary. For the selected sentences, we also ask the annotators to label its category: (1) the sentence is a general description of the topic; (2) the sentence is trying to explain why the topic is trending; (3) it is hard to tell the difference. Overall, the two annotators have high agreement (83.50%) regarding whether or not to include a sentence in the summary; among the selected summary sentences, only 22.58% of them were assigned with conflict purpose tags such as (1) or (2). Since some reference descriptions are simply repetition of others with very minor changes, we reduce the duplicates by iteratively removing the oldest sentences if all the consisting words are covered by the remaining sentence collection, until no sentence can be removed. All remaining sentences were concatenated to form one reference topic summary. On average, the reference summary for general and hashtag topics contains 44 and 40 words respectively.

7.2 Text Normalization[†]

The text messages serve as very valuable information sources, yet the nonstandard contents within them often degrade the existing language processing systems, calling the need of text normalization before applying the traditional information extraction, retrieval, sentiment analysis [98], or summarization techniques. Text message normalization is also of crucial importance for building robust and flexible text-to-speech (TTS) systems, which analyze the input noisy text messages and synthesize them into clear and natural speech without bringing any confusion to the users.

Text message normalization aims to replace the non-standard tokens that carry significant meanings with the context-appropriate standard English words. This is a very challenging task

[†]Work was done when Fei Liu was working as a research intern in the Research & Technology Center, Robert Bosch LLC, supervised by senior manager Fuliang Weng.

Table 7.2. Nonstandard tokens originated from the dictionary word "together" and their frequencies in the Twitter corpus.

2gether (6326)	togetha (919)	tgthr (250)	togeda (20)
2getha (1266)	togather (207)	t0gether (57)	toqethaa (10)
2gthr (178)	togehter (94)	togeter (49)	2getter (10)
2qetha (46)	togethor (29)	tagether (18)	2gtr (6)

due to the vast amount and wide variety of existing nonstandard tokens. We found more than 4 million distinct out-of-vocabulary tokens in the English tweets of the Edinburgh corpus (see Section 7.2.2). Table 7.2 shows examples of nonstandard tokens originated from the word "together". We can see that some variants can be generated by dropping letters from the original word ("tgthr") or substituting letters with digit ("2gether"); however, many variants are generated by combining the letter insertion, deletion, and substitution operations ("toqethaa", "2gthr"). This shows that it is difficult to divide the nonstandard tokens into exclusive categories.

Among the literature of normalizing real world text or text messages, [99, 100] employed the noisy channel model to find the most probable word sequence given the observed noisy message. Their approaches first classified the nonstandard tokens into various categories (e.g., abbreviation, stylistic variation, prefix-clipping), then calculated the posterior probability of the nonstandard tokens based on each category. [101] developed a hidden Markov word model using hand annotated training data. [102, 103] focused on modeling word abbreviations formed by dropping characters from the original word. [104] addressed the phonetic substitutions by extending the initial letter-to-phone model. [105, 106] viewed the text message normalization as a statistical machine translation process from the texting language to standard English. [107] experimented with the weighted finite-state machines for normalizing French SMS messages. Most of the above

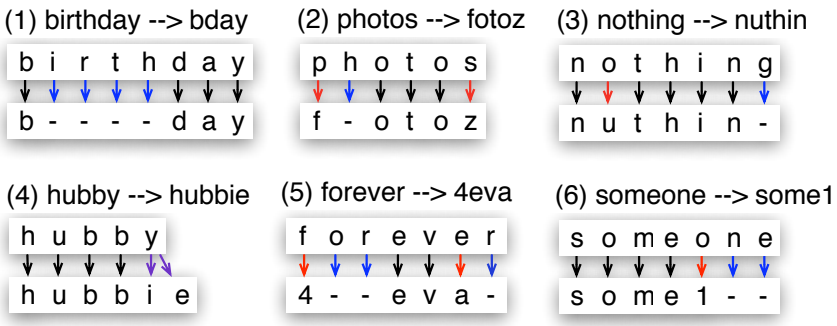


Figure 7.1. Examples of nonstandard tokens generated by performing letter transformation on the dictionary words.

approaches rely heavily on the hand annotated data and involve categorizing the nonstandard tokens in the first place, which gives rise to three problems: (1) the labeled data is very expensive and time consuming to obtain; (2) it is hard to establish a standard taxonomy for categorizing the tokens found in text messages; (3) the lack of optimized way to integrate various category-specific models often compromises the system performance, as confirmed by [100].

In this section, we propose a general letter transformation approach that normalizes nonstandard tokens without categorizing them. A large set of noisy training word pairs were automatically collected via a novel web-based approach and aligned at the character level for model training. The system was tested on both Twitter and SMS messages. Results show that our system significantly outperformed the jazzy spell checker and the state-of-the-art deletion-based abbreviation system, and also demonstrated good cross-domain portability.

7.2.1 General Framework

Given a noisy text message T, our goal is to normalize it into a standard English word sequence S. Under the noisy channel model, this is equivalent to finding the sequence $\hat{S}$ that maximizes

$p(S|T)$:

$$\hat{S} = \arg\max_S p(S|T) = \arg\max_S (\prod_i p(T_i|S_i))p(S)$$

where we assume that each non-standard token T_i is dependent on only one English word S_i, that is, we are not considering acronyms (e.g., "bbl" for "be back later") in this study. $p(S)$ can be calculated using a language model (LM). We formulate the process of generating a nonstandard token T_i from dictionary word S_i using a letter transformation model, and use the model confidence as the probability $p(T_i|S_i)$. Figure 7.1 shows several example (word, token) pairs[‡]. This transformation process will be learned automatically through a sequence labeling framework. To form a nonstandard token, each letter in the dictionary word can be labeled with: (a) one of the 0-9 digits; (b) one of the 26 characters including itself; (c) the null character "-"; (d) a letter combination. We integrate character-, phonetic-, and syllable-level features in the model that can effectively characterize the formation process of nonstandard tokens.

In general, the letter transformation approach will handle the nonstandard tokens listed in Table 7.3 yet without explicitly categorizing them. Note that the tokens with letter repetition are handled by first generating a set of variants by varying the repetitive letters (e.g. C_i = {"pleas", "pleeas", "pleaas", "pleeaas", 'pleeeaas"} for T_i = {"pleeeaas"}), then select the maximum posterior probability among all the variants:

$$p(T_i|S_i) = \max_{\tilde{T}_i \in C_i} p(\tilde{T}_i|S_i)$$

[‡]The ideal transform for example (5) would be "for" to "4". But in this study we are treating each letter in the English word separately and not considering the phase-level transformation.

Table 7.3. Nonstandard tokens that can be processed by the unified letter transformation approach.

(1) abbreviation	tgthr, weeknd, shudnt
(2) phonetic sub w/- or w/o digit	4got, sumbody, kulture
(3) graphemic sub w/- or w/o digit	t0gether, h3r3, 5top, doinq
(4) typographic error	thimg, macam
(5) stylistic variation	betta, hubbie, cutie
(6) letter repetition	pleeeaas, togtherrr
(7) any combination of (1) to (6)	luvvvin, 2moro, m0rnin

7.2.2 Web-based Data Collection w/o Supervision

One advantage of our proposed system is its robust performance using the automatically collected noisy training data, therefore avoiding the expensive human supervision. We use the Edinburgh Twitter corpus [108] for data collection, which contains 97 million Twitter messages. The English tweets were extracted using the TextCat language identification toolkit [109], and tokenized into a sequence of clean tokens consisting of letters, digits, and apostrophe using the Mallet toolkit [110].

For the out-of-vocabulary (OOV) tokens consisting of letters and apostrophe, we form 6 Google queries for each of them in the form of "w_1 w_2 w_3" OOV or OOV "w_1 w_2 w_3", where w_1 to w_3 are context words extracted from the tweets that contain this OOV. The first 32 returned snippets for each query were parsed and the bolded words with high frequency and common character sequence with the OOV were collected as possible standard word candidates. For the OOV tokens consisting of both letters and digits, we use simple rules to recover possible original words. These rules include: 1 → "one", "won", "i"; 2 → "to", "two", "too"; 3 → "e"; 4 → "for", "fore", "four"; 5 → "s"; 6 → "b"; 8 → "ate", "ait", "eat", "eate", "ight", "aight". The OOV token and any

resulting words were included in the noisy training pairs.

These noisy training pairs were further expanded and purged. We apply the transitive rule on these initially collected training pairs, so the two pairs "(cause, cauz)" and "(cauz, coz)" will incur "(cause, coz)" being added as another training pair. We remove the data pairs whose word candidate is not in the CMU dictionary, and those pairs for which word candidate and OOV are simply inflections of each other, e.g., "(headed, heading)". In total, we harvested 62,907 noisy training word pairs including 20,880 unique candidate words and 46,356 unique OOVs.

7.2.3 Letter Alignment

Given a training pair (S_i, T_i) consisting of a dictionary word S_i and its nonstandard variant T_i, we propose a two-pass procedure to align each letter in S_i with zero, one, or more letters/digits in T_i.

Pass (1): we first align the letters of the longest common sequence between the dictionary word and the variant, then align the letter chunks in between each of the existing alignments based on the following three cases:

- (many-to-0) If a chunk in the dict word needs to be aligned to zero letter in the variant, we map each letter in the chunk to "-" (e.g., "birthday" to "bday").
- (0-to-many) If zero letter in the dict word needs to be aligned to a letter/digit chunk in the variant, we check if the first letter in the chunk can be combined with the adjacent alignment to form a digraph (such as "wh", "ie"), e.g., "sandwich" aligned to "sandwhich". If not, we append the chunk to the next aligned variant letter.

- (many-to-many) Similarly, we first check if the first letter in the variant chunk can form a digraph with the adjacent alignment. If not, we map the chunk in the dict word to the chunk in the variant as one alignment, e.g., “someone” aligned to “some1”.

Pass (2): we remove the data pairs with chunk alignments involving more than three letters in either dict word or the variant. This is to eliminate possible noisy training pairs, such as (“virtualized”, “virtualization”). For the rest chunk alignments, we sequentially align the letters (e.g., “photos” aligned to “fotoz”). Note that for those 1-to-2 alignments, we align the single letter in the dict word to a two letter combination in the variant. We limit to the top 5 most frequent letter combinations, which are “ck”, “ey”, “ie”, “ou”, “wh”. After applying the letter alignment to the collected noisy training word pairs, we obtained 298,160 letter-level alignments. Some example alignments and corresponding word pairs are:

e → ’_’ (have, hav) q → k (iraq, irak)

e → a (another, anotha) q → g (iraq, irag)

e → 3 (online, 0nlin3) w → wh (watch, whatch)

7.2.4 Feature Extraction

We investigate a wide range of feature functions for effectively characterizing the letter transformation process. These features were then fed to a conditional random fields (CRF) model [111, 112] with L-BFGS for optimization. The features we used are:

- Character-level features

  Character n-grams: c_{-1}, c_0, c_1, $(c_{-2}\, c_{-1})$, $(c_{-1}\, c_0)$, $(c_0\, c_1)$, $(c_1\, c_2)$, $(c_{-3}\, c_{-2}\, c_{-1})$, $(c_{-2}\, c_{-1}\, c_0)$,

$(c_{-1}\ c_0\ c_1)$, $(c_0\ c_1\ c_2)$, $(c_1\ c_2\ c_3)$.

The relative position of the character in the word.

- Phonetic-level features

  Phoneme n-grams: p_{-1}, p_0, p_1, $(p_{-1}\ p_0)$, $(p_0\ p_1)$. We use the many-to-many letter-phoneme alignment algorithm [113] to map each letter to multiple phonemes (1-to-2 alignment). We use three binary features to indicate whether the current, previous, or next character is a vowel.

- Syllable-level features

  Relative position of the current syllable in the word; two binary features indicating whether the character is at the beginning or the end of the current syllable. The English hyphenation dictionary [114] is used to mark all the syllable information.

7.2.5 Normalization Results

We evaluate the normalization performance on both Twitter and SMS message test sets. The SMS test set was used in previous work [101, 100], which consists of 303 distinct nonstandard tokens and their corresponding dictionary words. We developed our own Twitter message test set consisting of 6,150 manually annotated tweets conducted via the Amazon Mechanical Turk. 3 to 6 turkers were required to normalize each tweet and convert the nonstandard tokens into standard English words. To facilitate annotation, we provide up to three candidates for each token (generated mainly based on spell checker, and including the original token). The turkers can also input the correct standard English words in case none of the candidates is correct. We extract the nonstandard tokens whose most frequently normalized word form consists of letters/digits/apostrophe, and is

Table 7.4. System accuracies using different configurations and n-best output.

System Accuracy	**Twitter** (**3802** pairs)		**SMS** (**303** pairs)	
	1-best	3-best	1-best	3-best
InternetSlang	7.94	8.07	4.95	4.95
(Pennell et al. 2010)	20.02	27.09	21.12	28.05
Jazzy Spell Checker	47.19	56.92	43.89	55.45
LetterTran (Trim)	**57.44**	**64.89**	**58.09**	**70.63**
LetterTran (All)	**59.15**	**67.02**	**58.09**	**70.96**
LetterTran (All) + Jazzy	**68.88**	**78.27**	**62.05**	**75.91**
(Choudhury et al. 2007)	n/a	n/a	59.9	n/a
(Cook et al. 2009)	n/a	n/a	59.4	n/a

different from the token itself. This results in 3,802 distinct nonstandard tokens that we use as the test set. 147 (3.87%) of them have more than one corresponding standard English words. Similar to prior work, we use isolated non-standard tokens without any context, that is, the LM probabilities $P(S)$ are based on unigrams.

We compare our system against three approaches. The first is a comprehensive list of chat slangs, abbreviations, and acronyms collected by InternetSlang.com; it contains normalized word forms for 6,105 commonly used slangs. The second is the word-abbreviation lookup table generated by the supervised deletion-based abbreviation approach proposed in [103][§]. It contains 477,941 (word, abbreviation) pairs automatically generated for 54,594 CMU dictionary words. The third is the jazzy spell checker based on the Aspell algorithm [115]. It integrates the phonetic matching algorithm (DoubleMetaphone) and Ispell's near miss strategy that enables the interchanging of two adjacent letters, and changing/deleting/adding of letters. The system performance is

[§]Thank Deana Pennell for sharing the look-up table generated for the CMU dictionary using the deletion-based abbreviation model.

measured using the n-best accuracy (n=1,3). For each nonstandard token, the system is considered correct if any of the corresponding standard words is among the n-best output from the system.

Results of system accuracies are shown in Table 7.4. For the system "LetterTran (All)", we first generate a lookup table by applying the trained CRF model to the CMU dictionary to generate up to 30 variants for each dictionary word[¶]. To make the comparison more meaningful, we also trim our lookup table to the same size as the deletion table, namely "LetterTran (Trim)". The trimming was performed by selecting the most frequent dictionary words and their generated variants until the length limit is reached. Word frequency information was obtained from the entire Edinburgh corpus. For both the deletion and letter transformation lookup tables, we generate a ranked list of candidate words for each nonstandard token, by sorting the probability $p(T_i|S_i) \times p(S_i)$, where $p(T_i|S_i)$ is the model confidence and $p(S_i)$ is the unigram count generated from the Edinburgh corpus. Since the string similarity and letter switching algorithms implemented in jazzy can compensate the letter transformation model, we also investigate combining the two ranked outputs "LetterTran(All) + Jazzy". In this configuration, we combine the candidate words from both systems and rerank them according to the unigram frequency; since the "LetterTran" itself is very effective in ranking candidate words, we only combine the jazzy output for tokens that "LetterTran" is not very confident about its best candidate (($p(T_i|S_i) \times p(S_i)$ is less than a threshold $\theta = 100$).

We notice the accuracy using the InternetSlang list is very poor, indicating text message normalization is a very challenging task that can hardly be tackled by using a hand-crafted list. The deletion table has a modest performance given the fact that it covers only deletion-based abbrevi-

[¶]We heuristically choose this large number since the learned letter/digit insertion, substitution, and deletion patterns tend to generate many variants for each dictionary word.

ations and letter repetitions (see Section 7.2.1). The "LetterTran" approach significantly outperforms all baselines even after trimming. This is because it inherently handles different ways of forming nonstandard tokens in an unified framework. Taking the Twitter test set for an example, the lookup table generated by "LetterTran" covered 69.94% of the total test tokens, among them, 96% were correctly normalized in the 3-best output, resulting in 67.02% overall accuracy. The test tokens that were not covered by the "LetterTrans" model include those generated by accidentally switching and inserting letters (e.g., "absolotuely" for "absolutely") and slangs ("addy" or "address"). Adding the output from jazzy compensates these problems and boosts the 1-best accuracy, achieved 21.69% and 18.16% absolute performance gain respectively on the Twitter and SMS test sets, as compared to using jazzy only. We also observe that the "LetterTran" model can be easily ported to the SMS domain. When combined with jazzy module, it achieved 62.05% 1-best accuracy, outperforming the domain-specific supervised system reported in [101] (59.9%) and the pre-categorized approach by [100] (59.4%). Regarding different feature categories, we found the character-level features are strong indicators, and using phonetic- and syllabic-level features also slightly benefits the performance.

7.3 Twitter Topic Summarization

For each of the topic phrases, our goal is to generate a short textual summary that can best convey the main ideas of the topic contents. We explore and compare multiple text sources as summarization input, including the user-contributed tweets, web contents linked from the tweets, as well as combination of the two sources. The concept-based optimization approach (as in [28, 29]) was employed for selecting informative summary sentences and minimizing the redundancy.

Note that our focus of this word is not developing new summarization systems, but rather utilizing and integrating different text sources for generating more informative Twitter topic summaries.

7.3.1 Concept-based Optimization Framework

Concept-based summarization approach first extracts a set of important concepts for each topic, then selects a collection of sentences that can cover as many important concepts as possible, while within the specified length limit. This idea is realized using the integer linear programming-based (ILP) optimization framework, with objective function set to maximize sum of the weighted concepts:

$$\max \quad \sum_i w_i c_i$$

where c_i is a binary variable indicating whether the concept i was covered by the summary; w_i is the weight associated to c_i.

We enforce two sets of length constraints to the summary: sentence- or word-based. Sentence constraint requires the total number of selected summary sentences to not exceed a length limit L_1; while word constraint requires the total words of selected sentences not exceed length limit L_2; s_j is a binary variable indicating whether sentence j was selected in the summary; l_j represents the number of words in s_j.

$$\sum_j s_j < L_1 \quad or \quad \sum_j l_j s_j < L_2$$

Further, we connect concept c_i with sentence s_j using two sets of constraints. The binary variable o_{ij} is used to indicate whether concept c_i exists in sentence s_j. For all the sentences that contain concept c_i, if any sentence was selected in the summary, the concept c_i should be

covered by the summary; reversely, if c_i was covered by the summary, at least one of the sentences containing c_i should be selected.

$$\forall i \quad c_i \leq \sum_j o_{ij} s_j$$

$$\forall i, j \quad c_i \geq o_{ij} s_j$$

The concepts are selected by extracting n-grams (n=1, 2, 3) from the input documents corresponding to each topic. Similar to [29], we remove (1) n-grams that appear only once in the documents; (2) n-grams that have a consisting word with inverse document frequency (IDF) value lower than a threshold; (3) n-grams that are enclosed by higher order n-grams with the same frequency. These filters are designed to exclude insignificant n-grams from the concept set. The IDF scores were calculated from a large background corpus corresponding to the input text source, using individual sentence or tweet as a pseudo-document; words with low IDF scores (such as stopwords) tend to appear in many sentences and therefore should be removed from the concept set. We assign a weight to each n-gram concept as follows:

$$w_i = tf(ngram_i) \times n \times \max_j \ idf(w_{ij})$$

where w_i is the concept weight, $tf(ngram_i)$ is the term frequency of $ngram_i$ in the input documents of the topic; n denotes the order of $ngram_i$; w_{ij} are the consisting words of $ngram_i$; $idf(w_{ij})$ represents IDF value of word w_{ij}. This approach aims to extract n-grams that appear frequently in each topic, but do not appear frequently in a large background corpus. The weights are also biased towards longer n-grams since they carry more information.

7.3.2 Summarization Input

In this section, we explore different text sources as input to the summarization system. Different from previous studies that take input from a single text source, we propose to utilizing both the user-contributed tweets and the linked web contents for Twitter topic summarization, since these two sources provide very different text quality and may contain complementary information regarding the topic. These text sources also pose great challenges to the summarization system: the tweets are short and extremely noisy; while the web pages linked from the tweets may have vast different layouts and contain a variety of information.

Original Tweets

As shown in Table 7.1, the initially collected tweets are very noisy. They are passed through a set of preprocessors to remove non-ascii characters, HTML special characters, URLs, emoticons, punctuation marks, retweet tags (RT @user), etc. We also remove the reply (@) and hashtag (#) tokens that do not carry syntactic roles (such as in the subject or object position) by using a set of regular expressions. These preprocessed tweets are sorted by date and taken as the first input source to the summarization system (denoted by "OrigTweets").

Normalized Tweets

The original tweets contain various nonstandard word tokens, such as the ones listed in Table 7.3. They can be originated from the users' need of fast typing, twitter's 140-character limit, various input devices, or related to the socialinguistic phenomena [116]. We hypothesize that by

normalizing these nonstandard tokens into standard English words and using the normalized tweets as input can help boost the summarization performance.

We utilize the proposed twitter message normalization system in Section 7.2. We identify the nonstandard tokens that need to be normalized using the following criteria: (1) it is not in the CMU dictionary; (2) it does not contain capitalized letter; (3) it appears infrequently in the topic (less than a threshold); (4) it is not a popular chat acronyms (such as "lol", "omg"); (5) it contains letters/digits/apostrophe, but should not be numbers only. These criteria are designed to avoid normalizing the named entities, frequently appearing out-of-vocabulary terms (such as "itunes"), chat acronyms, usernames, and hashtags. The selected nonstandard tokens in the original tweets will be replaced by the system generated 1-best candidate word. Note that we do not discriminate the context when replacing each nonstandard token. This will be addressed in the future work. We use these normalized tweets as a second source of summarization input and name them "NormTweets".

Linked Web Contents

For each Twitter topic, we collect a set of web pages linked by the topic tweets and use them as another source of summarization input. An example linked web page was demonstrated in Table 7.1. To collect the web pages, we first identify the URLs from all tweets, then for each topic select up to n (n = 10) URLs that appear most frequently in the topic tweets and infrequently across different Twitter topics. This scheme is similar to the TF-IDF measure. This way we can select the salient URLs for each topic while avoiding the spam URLs. The contents of these URLs were

collected and only distinct web pages were retained. We use an HTML parser[‖] to clean the web pages and collect the textual contents. We also performed sentence segmentation[**] on the parsed web pages. All the pages corresponding to the same topic were sorted by the date they were first cited in the tweets. These web pages were taken as another input text source for the summarization system, denoted as "Web".

Combining Tweets and Web Contents

We expect that taking advantage of both tweets and linked web contents would benefit the topic summarization system. Consolidating the distinct text sources may help boost the weight of key concepts and eliminate the spam information. As a preliminary study, we investigate concatenating either the original tweets or the normalized tweets with the linked web pages as input to the concept-based summarization system. This results in two inputs "Web + OrigTweets" and "Web + NormTweets". We will explore other ways of combining the two text sources in future work.

7.4 Experiments

7.4.1 Experimental Setup

Among all the collected topics, we select 500 general topics (such as "Chilean miners") and 50 hashtag topics (such as "#octoberwish", "#wheniwasakid") for experimentation. On average, a general topic contains 1673 tweets and 3.43 extracted linked web pages; while a hashtag topic contains 3316 tweets but does not have meaningful linked web pages.

[‖] http://jericho.htmlparser.net/docs/index.html

[**] http://www.cs.umd.edu/Honors/reports/Elkis-honorspaper/node6.html

The concept-based optimization system was configured to extract a collection of sentences/tweets for each topic, using either the sentence- or word-constraint (denoted as "#Sent" and "#Word"). We opt to set individual length constraint for each topic rather than using a uniform length limit for all topics, since the topics can be very different in length and duration. We use the number of sentences/words in the reference summary as the sentence/word constraint for each topic. Note that in practice this reference summary length information may not be available. We use the length constraints obtained from the reference summary in this exploratory study, since our focus is to first evaluate if twitter trending summarization is feasible, and what are the effects of different information sources and non-standard tokens. The ROUGE-1 F-scores [63] are used to measure the n-gram (n=1) overlap between the system summaries and reference summaries. We also performed human evaluation by asking annotators to score both the system and reference summaries regarding the linguistic quality and content responsiveness, in the hope this will benefit future research in this direction.

7.4.2 Automatic Evaluation Results

General Topics		R-1 F(%)		RefSum
Input Source	Render	#Sent	#Word	Cov(%)
OrigTweets	Orig	29.53	30.21	94.81
	Norm	29.41	30.21	94.81
NormTweets	Norm	29.69	30.35	94.60
Web		24.32	25.07	63.74
Web + OrigTweets		29.58	30.44	95.37
Web + NormTweets		29.66	**30.54**	95.16

Table 7.5. ROUGE-1 F-measure and reference summary coverage scores for general topics.

We present the summarization results (ROUGE-1 F-measure) for the general topics in Table 7.5. Five different text sources were exploited as the system inputs, as described in Section 7.3.2. To measure the quality of the input for summarization, we also include reference summary coverage score in the table, defined as the percentage of words in the reference summary that are covered by the input text source. When using tweets as input, we also investigate whether we should apply tweet normalization before or after the summarization process, that is "pre-normalization" (using "NormTweets" as input), or "post-normalization" (using "OrigTweets" as input, and rendering the normalized summary tweets).

In general, we notice the ROUGE scores are lower compared to summarization in other text domains. This indicates that Twitter topic summarization is very challenging. Comparing the two constraints used in the concept-based optimization framework, we found that the word constraint performs constantly better for the general topics. This is natural since the word constraint tightly bounds the length of the system output, while the sentence constraint is relatively loose. For the different sources, we notice using linked web pages alone yields worse summarization performance, as well as lower reference summary coverage; however, when combined with the tweets, there is a slight increase in the coverage scores, and sometimes summarization results. This suggests that the linked web pages can contain extra useful information for generating summaries. Regarding normalization, results show that the "pre-normalization" (using normalized tweets as input) can generally improve the summary tweet selection. For general topics, the best performance was achieved by combining the normalized tweets and linked web pages as input source and using the word-level constraint.

Results for hashtag topics were shown in Table 7.6 using tweets as input (there are no linked

Hashtag Topics		R-1 F(%)		RefSum
Input Source	Render	#Sent	#Word	Cov(%)
OrigTweets	Orig	9.08	7.19	93.93
	Norm	9.09	7.16	93.93
NormTweets	Norm	**9.35**	7.14	93.71

Table 7.6. ROUGE-1 F-measure and reference summary coverage scores for hashtag topics.

webpages for these topics). We notice the reference coverage scores are satisfying, yet the system output barely matches the reference summaries (very low ROUGE-1 scores). Looking at the reference and system generated summaries for the hashtag topics, we found the system output is more specific (e.g., "#octoberwish everything goes well."), while the reference summaries are often very general (e.g., "people tweeting about their wishes for October."). The human annotators also noted that most hashtag topics (such as "#octoberwish", "#wheniwasakid") are self-explainable and may require special attention to redefine an appropriate summary. Using sentence constraints yield better performance than word-based one, with larger performance difference than that for the general topics. For hashtag topics, the best performance was achieved using the "pre-normalization" with sentence constraint.

For an analysis, we generate oracle system performance by using the reference summaries to extract a set of unweighted concepts to use in the ILP optimization framework for sentences/tweets selection. This results in 61.76% ROUGE-1 F-score for the general topics and 40.34% for the hashtag topics, indicating abundant space for future improvement. About the effect of normalization, in most cases it helps to use normalized tweets. We found the normalization system replaced 1.08% and 1.8% of the total word tokens for the general and hashtag topics respectively; these tokens spread in 13.12% and 16.85% of the total tweets.

7.4.3 Human Evaluation Results

	General			Hashtag	
	Tweet	Web	Ref	Tweet	Ref
Gram.	3.13	**3.42**	4.52	3.04	4.24
NRedun.	3.93	**4.64**	4.30	4.82	3.62
Clarity	**4.07**	3.91	4.77	4.06	4.60
Focus	**3.64**	3.03	4.75	3.22	4.72
Content	2.82	2.55	n/a	2.60	n/a
ExtraInfo	n/a	2.63	n/a	n/a	n/a

Table 7.7. Linguistic quality, content coverage, and usefulness scores judged by human assessors.

We ask two human annotators to manually evaluate the system and reference summaries regarding the readability and content coverage. Readability includes grammaticality, non-redundancy, referential clarity, and focus; content coverage was evaluated for system summaries against the reference summary. The annotators were also asked to rate the "Web" summaries regarding whether they provided extra useful topic information on top of the "Tweet" summary. 50 general topics and 25 hashtag topics were randomly selected for assessment. The "Tweet" and "Web" summaries were generated using the original tweets and linked web pages with word constraint for general topics, and sentence constraint for hashtag topics. Each of the assessors was asked to judge all the summaries and assign a score for each criterion on a 1 to 5 Likert scale (5 being the best quality). The average scores of the two assessors were presented in Table 7.7.

For general topics, the "Web" summaries outperform the "Tweet" summaries on both grammaticality and non-redundancy, confirming the advantage of using the high-quality linked web pages. The referential clarity and focus scores of the "Web" summaries are not very high, since the summary sentences were extracted simultaneously from several web pages, and the system subjects

to similar challenges as in multi-document summarization. The content coverage scores of both system summaries seem to correlate well with the ROUGE-1 F-measure, with a higher score for "Tweet" summaries. The assessors also rate that 48% of the "Web" summaries contain "Somewhat Useful" extra topic information, and 21% are"Very Useful". Note that this could be just because of the inherent difference of the two summaries, regardless of the input source, but in general we believe the linked web pages (such as the news documents) can provide more detailed and coherent stories as compared to the 140-character tweets. For hashtag topics, the "Tweet" summaries yield worse grammaticality and focus scores, but have very high non-redundancy score. On the contrary, the reference summaries often contain redundant information. The content match score between the system and reference summaries (2.6) does not seem to reflect the ROUGE scores. We hypothesize that even though the specificity of the two summaries is different, the assessors may still think the system summaries match the reference ones to some extent. A larger scale human evaluation is needed to study the correlation between human and automatic evaluation.

We show an example of reference and system generated summaries for a general and a hashtag topic in Table 7.8, and summarize some challenges for this summarization task below:

- **Reference summaries are hard to create.** The reference descriptions from WhatTheTrend.com were created by Twitter users, which would be unavoidably biased to the information available in Twitter. The user-contributed descriptions may also contain spam descriptions, repetitions, nonstandard tokens, etc. It would be better to have a concise non-redundant sentence collection for developing future summarization systems. In particular, hashtag topics need special attention. They account for 40% of the total trending topics in 2010 according to the

Table 7.8. Example system and reference summaries for both general and hashtag topics.

General Topic: “3PAR”	
RefSum	Dell Inc. and Hewlett-Packard Co. are both bidding for storage device maker 3Par Inc. 3Par jumped 21 percent after Hewlett- Packard Co. offered $30 a share for the company.
TweetSum	Dell ups 3Par offer yet again, to $27 per share Dell Raises 3par Offer to Match HP Bid Dell Matches HP’s Offer for 3Par, Boosting Bid to $1.8 Billion
WebSum	Dell Matches HP’s $27 Offer, Is Accepted by 3PAR. 3PAR has accepted an increased acquisition offer from Dell of US$27 per share, matching Hewlett-Packard’s earlier raised bid.
Hashtag Topic: “#wheniwasakid”	
RefSum	when i was a kid.... people are sharing there best (good or bad) memories from childhood. People reminise the wonderful times about being a kid.
TweetSum	#whenIwasakid getting wasted meant eating all the ice cream and candy you could until you puked! #whenIWasAKid Apple & Blackberry were fruits not phones.

statistics in WhatTheTrend.com[††]. Yet there still lacks standard definition regarding a good hashtag summary. From the example topic "#wheniwasakid" in Table 7.8, we can see they are very different in nature from general topics, thus future efforts are needed to define an appropriate summary.

- **Evaluation issues.** Word based evaluation measures will rarely consider semantic relatedness between concepts, or name entity variations, such as "Hewlett-Packard" vs. "HP", "Dell ups 3Par offer" vs. "Dell Raises 3par Offer", etc. When comparing the system summaries with short human-written reference summaries, the word overlap varies a lot for different human summarizers.

- **Dynamically changing topics/events.** Some general topics are related to events that are constantly changing. Take the "3PAR" topic in Table 7.8 for an example, two companies take turns to raise the bid for 3Par Inc., a good topic summary should be able to develop a series of sub-events and show the topic evolving process.

7.5 Summary and Discussions

In this chapter, we proposed to explore a variety of text sources for summarizing the Twitter topics, including the user-contributed tweets, normalized tweets, linked web contents, as well as the combination of different text sources. We proposed a letter transformation approach for general text message normalization without pre-categorizing the nonstandard tokens into different types (e.g., insertion, deletion, substitution). We also avoided the expensive and time consuming hand labeling process by automatically collecting and aligning a large set of noisy training

[††]http://yearinreview.whatthetrend.com/

pairs. Results on Twitter and SMS domain show that our system can significantly outperform the state-of-the-art systems and have good domain portability. For twitter topic summarization, we employed the concept-based optimization framework with multiple input text sources to generate the summaries. We conducted both automatic and human evaluation regarding the summary quality. Better performance is observed when using the normalized tweets as input, indicating special treatment should be performed before feeding the noisy tweets to the summarization system. We also found the linked web contents can provide extra useful topic information. In future work, we will compare our system with other dedicated microblog summarization systems, as well as address some of the challenges identified in this study.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

In this work, we proposed to extract keywords using a novel supervised framework that incorporates various knowledge sources: beyond the traditionally widely used features (e.g., TF-IDF, position information), we introduce additional rich features including term specificity information, decision-making sentence related features, speaker and prominence based features, and features extracted from system generated summaries. We propose a feedback strategy to reinforce the impact of summary sentences on selecting effective keywords. We conduct analysis to evaluate feature effectiveness using different feature selection processes, and define various measurements to characterize the quality of summaries that can benefit the keyword extraction task. We also evaluate system performance using both human transcripts and different ASR output (1-best and n-best), and show promising improved keyword extraction results using n-best ASR output over 1-best hypotheses.

For extractive meeting summarization, we explore multiple meeting-specific characteristics. We propose to use agenda and speaker-dependent characteristics (such as verboseness, gender, native language, role in the meeting) to improve extractive meeting summarization performance. These properties were incorporated in both unsupervised Maximum Marginal Relevance (MMR) approach and the supervised framework. We observe consistent improvements using our proposed

approaches, on both human transcripts and ASR output, and using different evaluation metrics including ROUGE, Pyramid, and a DA-level F-measure score. Beyond extractive summarization, we propose to perform sentence compression on the extractive summary to improve its readability and make it more like an abstractive summary. Various automatic compression algorithms were investigated, including the integer linear programming (ILP) based approach with filler phrase detection, a noisy-channel approach using Markovization formulation of grammar rules, as well as the conditional random fields (CRF) based approach. The automatically compressed utterances were compared against both human compression and the abstractive summaries. We also evaluate the impact of using compressed utterances on summarization, and propose a fully automatic summarizer that generates compressed meeting summaries by combing the utterance compression module with an extractive summarization system.

We perform exploratory keyword extraction and summarization on a similar domain of conversational text, to help the Twitter users quickly browse through any available topics. As an important step of preprocessing, we propose a novel letter transformation approach to convert the nonstandard tokens in the Twitter posts into standard English words. Different from prior work, our approach requires neither pre-categorization nor human supervision. It models the generation process from the dictionary words to nonstandard tokens under a sequence labeling framework. We explore summarizing the Twitter topics using the concept-based global optimization approach, and investigate the potential impact of the noisy nonstandard tokens on the summarization performance.

8.2 Future Work

In this section, we propose a series of future research directions, including outlining the challenges and possible directions for extracting keywords from the meeting transcripts; performing automatic topic segmentation and topic-level meeting summarization; using utterance compression to assist meeting summarization and generate more condensed meeting summaries; as well as posing the future work on multi-faceted summarization of opinionated conversational text.

- Chapter 4 summarized several challenges regarding extracting keywords from meeting transcripts, including the typical frequency-based approaches, human annotation and evaluation issues, high word error rates, and the unigram based approaches. Among them, the low lexical density of meeting transcripts has notable side effect on the traditional frequency-based keyword extraction systems. In the future, we will investigate integrating the semantic relatedness information into the keyword extraction system, since the ICSI meeting corpus and many other meeting corpora often consist of a series of semantically related meeting conversations. For example, a set of semantically related concepts can be first extracted based on the entire corpus and expanded using external knowledge sources (such as Wikipedia); these concepts can be further used as a controlled vocabulary for assigning keywords to individual meetings, thus resulting in more coherent keyword sets.

- Each meeting in the ICSI corpus lasts about an hour, and has an average of 8 to 10 top-level topic segments. Each of these topic segments has a human labeled short description: "Digit task/Opening/Closing/Chitchat" are the functional labels, labels such as "hire Fey as wizard" denote on-topic discussion. In Chapter 5, we investigate using these topic labels as pseudo-

agenda items to improve the summarization performance. In [38], the authors show that the topic-level features are helpful for the meeting summarization task. In the future, we will explore the topic-based meeting summarization. The benefits of performing the topic-level summarization are two folds: first, it can help generate more coherent meeting summaries. Consider the general procedure of meeting discussion, an issue was first proposed by one of the speakers, then possible solutions were discussed among participants, finally a conclusion or decision was reached (or not). Knowing the general discussion flow can help the topic-level summarization systems build a more structured and coherent story. Second, users of a meeting browser might raise questions about a specific issue, such as "Was this issue discussed in last week's meeting?", "Was any conclusions reached with respect to this issue?", or "Is there any update for this issue in the following meetings?". Topic-level summaries can be very useful for these situations since they summarize individual issues/topics and are more appropriate for catering the specific user information need. The approaches for topic-level summarization could involve the automatic topic segmentation in the first stage, then perform summarization on each individual topic segment. The task of topic segmentation and summarization can also be modeled as an interactive process: automatic summarization module first picks up important sentences from the entire meeting, then employs a clustering scheme to group these summary sentences into topics, which can in turn be used to extract summary sentences from, and so on so forth. Summarization of each individual topic segment could be related to the sentences' functionality such as "problem proposing", "opinionated discussion", "general discussion", "decision making", etc. Ultimately, a structured and coherent summary toward the specific issue discussed in this topic segment can be expected.

- Speech utterance compression can help remove the fillers, editing terms, colloquial expressions, redundancies, etc. contained in the spoken utterances, therefore is expected to increase the readability of the meeting summaries generated by the current extractive approaches. Different from the abstractive summarization approaches that often require complicate domain ontology and language generation techniques, combining compression with summarization allows us to generate condensed meetings summaries by coupling two mature language processing techniques. In the future, we will investigate approaches to jointly optimize the summary sentence selection and compression, considering prosody information in the compression model, using syntactic parser to help prune the unnecessary subclauses, as well as evaluating the system performance on the automatically recognized meeting transcripts.

- Another interesting extension of the current work is to create multi-faceted summaries for the conversational text. During the discussion of a project or an event, every participant could raise his/her opinion regarding a specific facet of the event. For example, regarding the Japan earthquake, there are a variety of microblog posts commenting on the magnitude of the earthquake, the tsunami, the stricken Fukushima nuclear plant, the quake's aftermath, and support relief efforts. It will be very nice to combine the latent topic modeling approaches with the summarization system to automatically generate a multi-faceted summary for the entire event, with the number of summary sentences extracted in proportion to the size of the latent sub-topic. Moreover, the conversational texts are often opinionated. In meeting dialogues, this is mostly expressed via the speech prosody; while in other conversational texts, users have adopted many creative ways to emphasize things or express their emotions, including the use of repeating letters ("sooo", "yaay", "rescueeed"), multiple consecutive

punctuation marks (“!!!!!”), capitalizing the first letter or all letters of some words (“AMAZ-ING”), or using emoticons (“:-)”). These extra information sources will be considered in future work on top of the positive and negative opinions carried in the surface words.

REFERENCES

[1] Leena Rao, "Twitter seeing 90 million tweets per day, 25 percent contain links," *http://techcrunch.com/2010/09/14/twitter-seeing-90-million-tweets-per-day/*, 2010.

[2] Michael Alexander Kirkwood Halliday, "Some grammatical problems in scientific english," *Writing Science: Literacy and Discursive Power*, pp. 69–85, 1993.

[3] Shaomei Wu, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts, "Who says what to whom on twitter," in *Proc. of WWW*, 2011.

[4] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning, "Domain-specific keyphrase extraction," in *Proceedings of IJCAI*, 1999, pp. 668–673.

[5] Yaakov HaCohen-Kerner, "Automatic extraction of keywords from abstracts," in *Proceedings of the Seventh International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, 2003, vol. 2773, pp. 843–849.

[6] Yaakov HaCohen-Kerner, Zuriel Gross, and Asaf Masa, "Automatic extraction and learning of keyphrases from scientific articles," in *Computational Linguistics and Intelligent Text Processing*, 2005, pp. 657–669.

[7] Steve Jones and Gordon W. Paynter, "Automatic extraction of document keyphrases for use in digital libraries: Evaluation and applications," *Journal of the American Society for Information Science and Technology*, vol. 53, no. 8, pp. 653–677, 2002.

[8] Peter D. Turney, "Coherent keyphrase extraction via web mining," in *Proceedings of IJCAI*, 2003, pp. 434–439.

[9] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-

Manning, "KEA: Practical automatic keyphrase extraction," in *Proceedings of ACM Digital Libraries*, 1999, pp. 254–256.

[10] Kirill Kireyev, "Semantic-based estimation of term informativeness," in *Proceedings of NAACL*, 2009, pp. 530–538.

[11] Anette Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Proceedings of EMNLP*, 2003, pp. 216–223.

[12] Lonneke van der Plas, Vincenzo Pallotta, Martin Rajman, and Hatem Ghorbel, "Automatic keyword extraction from spoken text. a comparison of two lexical resources: the EDR and WordNet," in *Proceedings of LREC*, 2004, pp. 2205–2208.

[13] Xiaojun Wan, Jianwu Yang, and Jianguo Xiao, "Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction," in *Proceedings of ACL*, 2007, pp. 552–559.

[14] Kino High Coursey, Rada Mihalcea, and William E. Moen, "Automatic keyword extraction for learning object repositories," in *Proceedings of the Conference of the American Society for Information Science and Technology*, 2008.

[15] Diana Inkpen and Alain Désilets, "Extracting semantically-coherent keyphrases from speech," *Canadian Acoustics Association*, vol. 32, pp. 130–131, 2004.

[16] Andras Csomai and Rada Mihalcea, "Linguistically motivated features for enhanced back-of-the-book indexing," in *Proceedings of ACL*, 2008, pp. 932–940.

[17] Yutaka Matsuo and Mitsuru Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information," *International Journal on Artificial Intelligence*, vol. 13, no. 1, pp. 157–169, 2004.

[18] Yaakov HaCohen-Kerner, Ittay Stern, David Korkus, and Erick Fredj, "Automatic machine learning of keyphrase extraction from short html documents written in hebrew," *Cybernetics and Systems*, vol. 38, no. 1, pp. 1–21, 2007.

[19] Peter D. Turney, "Learning algorithms for keyphrase extraction," *Information Retrieval*, vol. 2, no. 4, pp. 303–336, 2000.

[20] Hongyuan Zha, "Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering," in *Proceedings of SIGIR*, 2002, pp. 113–120.

[21] Ernesto D'Avanzo, Bernardo Magnini, and Alessandro Vallin, "Keyphrase extraction for summarization purposes: The LAKE system at DUC-2004," in *Proceedings of Document Understanding Conference*, 2004.

[22] Alain Désilets, Berry de Bruijn, and Joel Martin, "Extracting keyphrases from spoken audio documents," *Information Retrieval Techniques for Speech Applications*, vol. 2273, pp. 36–50, 2002.

[23] Anette Hulth, "Reducing false positives by expert combination in automatic keyword indexing," in *Proceedings of RANLP*, 2003, pp. 197–203.

[24] Klaus Zechner, "Automatic summarization of open-domain multiparty dialogues in diverse genres," *Computational Linguistics*, vol. 28, no. 4, pp. 447–485, 2002.

[25] Jaime Carbonell and Jade Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proc. of ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[26] Gabriel Murray, Steve Renals, Jean Carletta, and Johanna Moore, "Evaluating automatic summaries of meeting recordings," in *Proceedings of ACL 2005 MTSE Workshop*, 2005, pp. 39–52.

[27] Gabriel Murray, Steve Renals, Jean Carletta, and Johanna Moore, "Incorporating speaker and discourse features into speech summarization," in *Proc. of HLT-NAACL*, 2006.

[28] Dan Gillick, Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tür, "A global optimization framework for meeting summarization," in *Proc. of ICASSP*, 2009.

[29] Shasha Xie, Dilek Hakkani-Tür, Benoit Favre, and Yang Liu, "Integrating prosodic features in extractive meeting summarization," in *Proc. of ASRU*, 2009.

[30] Günes Erkan and Dragomir R. Radev, "LexRank: Graph-based centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, no. 1, pp. 457–479, 2004.

[31] Rada Mihalcea and Paul Tarau, "TextRank: Bringing order into texts.," in *Proceedings of EMNLP*, 2004, pp. 404–411.

[32] Xiaojun Wan, Jianwu Yang, and Jianguo Xiao, "Manifold-ranking based topic-focused multi-document summarization," in *Proc. of IJCAI*, 2007.

[33] Nikhil Garg, Benoit Favre, Korbinian Reidhammer, and Dilek Hakkani-Tür, "ClusterRank: A graph base method for meeting summarization," in *Proc. of Interspeech*, 2009.

[34] Sameer Maskey and Julia Hirschberg, "Summarizing speech without text using hidden Markov models," in *Proc. of HLT/NAACL*, 2006.

[35] Anne Hendrik Buist, Wessel Kraaij, and Stephan Raaijmakers, "Automatic summarization of meeting data: A feasibility study," in *Proc. of CLIN*, 2005.

[36] Michel Galley, "A skip-chain conditional random field for ranking meeting utterances by importance," in *Proc. of EMNLP*, 2006.

[37] Sameer Maskey and Julia Hirschberg, "Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization," in *Proc. of Eurospeech*, 2005.

[38] Shasha Xie, Yang Liu, and Hui Lin, "Evaluating the effectiveness of features and sampling in extractive meeting summarization," in *Proceedings of IEEE Workshop on Spoken Language Technology*, 2008, pp. 157–160.

[39] Jian Zhang, Ho Yin Chan, Pascale Fung, and Lu Cao, "A comparative study on speech summarization of broadcast news and lecture speech," in *Proc. of Interspeech*, 2007.

[40] Xiaodan Zhu and Gerald Penn, "Comparing the roles of textual, acoustic and spoken-language features on spontaneous-conversation summarization," in *Proc. of HLT-NAACL*, 2006.

[41] Xiaodan Zhu, Gerald Penn, and Frank Rudzicz, "Summarizing multiple spoken documents: Finding evidence from untranscribed audio," in *Proc. of ACL-IJCNLP*, 2009.

[42] Justin Jian Zhang, Ricky Ho Yin Chan, and Pascale Fung, "Extractive speech summarization using shallow rhetorical structure modeling," *IEEE Trans. on Audio, Speech, and Language Processing*, 2009.

[43] Shih-Hsiang Lin and Berlin Chen, "Improved speech summarization with multiple-hypothesis representations and Kullback-Leibler divergence measures," in *Proc. of Interspeech*, 2009.

[44] Shasha Xie and Yang Liu, "Using confusion networks for speech summarization," in *Proc. of NAACL*, 2010.

[45] Shih-Hsiang Lin and Berlin Chen, "A risk minimization framework for extractive speech summarization," in *Proc. of ACL*, 2010.

[46] Chiori Hori and Sadaoki Furui, "A new approach to automatic speech summarization," *IEEE Transactions on Multimedia*, vol. 5, no. 3, pp. 368–378, 2003.

[47] Gabriel Murray, Giuseppe Carenini, and Raymond Ng, "Interpretation and transformation for abstracting conversations," in *Proc. of NAACL*, 2010.

[48] Gabriel Murray, Giuseppe Carenini, and Raymond Ng, "The impact of ASR on abstractive vs. extractive meeting summaries," in *Proceedings of INTERSPEECH*, 2010.

[49] Gabriel Murray, Thomas Kleinbauer, Peter Poller, Tilman Becker, Steve Renals, and Jonathan Kilgour, "Extrinsic summarization evaluation: A decision audit task," *ACM Transactions on Speech and Language Processing*, vol. 6, no. 2, 2009.

[50] Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita, "Summarizing microblogs automatically," in *Proc. of HLT/NAACL*, 2010.

[51] Beaux Sharifi, Mark-Anthony Hutton, and Jugal K. Kalita, "Experiments in microblog summarization," in *Proc. of IEEE Second International Conference on Social Computing*, 2010.

[52] David Inouye, "Multiple post microblog summarization," *REU Research Final Report*, 2010.

[53] Brendan O'Connor, Michel Krieger, and David Ahn, "TweetMotif: Exploratory search and topic summarization for twitter," in *Proc. of the International AAAI Conference on Weblogs and Social Media*, 2010.

[54] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller, "TwitInfo: Aggregating and visualizing microblogs for event exploration," in *Proc. of CHI*, 2011.

[55] Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters, "The ICSI meeting corpus," in *Proceedings of ICASSP*, 2003, pp. 364–367.

[56] Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey, "The ICSI meeting recorder dialog act (MRDA) corpus," in *Proceedings of SIGdial Workshop on Discourse and Dialogue*, 2004, pp. 97–100.

[57] Steve Renals, Thomas Hain, and Hervé Bourlard, "Recognition and understanding of meetings: The AMI and AMIDA projects," in *Proc. of ASRU*, 2007.

[58] Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing, "Discourse segmentation of multi-party conversation," in *Proceedings of ACL*, 2003, pp. 562–569.

[59] Fei Liu, Feifan Liu, and Yang Liu, "Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion," in *Proceedings of IEEE Workshop on Spoken Language Technology*, 2008, pp. 181–184.

[60] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais, "The vocabulary problem in human-system communication," *Communications of the ACM*, vol. 30, no. 11, pp. 964–971, 1987.

[61] Jean Carletta, "Assessing agreement on classification tasks: the Kappa statistic," *Computational Linguistics*, vol. 22, pp. 249–254, 1996.

[62] Inderjeet Mani, Gary Klein, David House, Lynette Hirschman, Therese Firmin, and Beth Sundheim, "SUMMAC: a text summarization evaluation," *Natural Language Engineering*, vol. 8, pp. 43–68, 2002.

[63] Chin-Yew Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proceedings of ACL Workshop on Text Summarization Branches Out*, 2004, pp. 74–81.

[64] Fei Liu and Yang Liu, "What are meeting summaries? An analysis of human extractive summaries in meeting corpus," in *Proceedings of SIGDial Workshop on Discourse and Dialogue*, 2008, pp. 80–83.

[65] Jamie Reilly and Jacob Kean, "Formal distinctiveness of high- and low- imageability nouns: Analyses and theoretical implications," *Cognitive Science*, vol. 31, no. 1, pp. 157–168, 2007.

[66] Pei-Yun Hsueh and Johanna Moore, "What decisions have you made: Automatic decision detection in conversational speech," in *Proceedings of NAACL-HLT*, 2007, pp. 25–32.

[67] Raquel Fernández, Matthew Frampton, Patrick Ehlen, Matthew Purver, and Stanley Peters, "Modelling and detecting decisions in multi-party dialogue," in *Proceedings of SIGDial Workshop on Discourse and Dialogue*, 2008, pp. 156–163.

[68] Inderjeet Mani, *Advances in Automatic Text Summarization*, MIT Press, Cambridge, MA, USA, 1999.

[69] Gerald Penn and Xiaodan Zhu, "A critical reassessment of evaluation baselines for speech summarization," in *Proceedings of ACL*, 2008, pp. 470–478.

[70] Je Hun Jeon and Yang Liu, "Automatic accent detection: Effect of base units and boundary information," in *Proceedings of INTERSPEECH*, 2009, pp. 180–183.

[71] Furu Wei, Wenjie Li, Qin Lu, and Yanxiang He, "Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization," in *Proceedings of SIGIR*, 2008, pp. 283–290.

[72] Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu, "Unsupervised approaches to automatic keyword extraction using meeting transcripts," in *Proceedings of HLT-NAACL*, 2009, pp. 620–628.

[73] Jahna Otterbacher, Günes Erkan, and Dragomir R. Radev, "Using random walks for question-focused sentence retrieval," in *Proceedings of HLT-EMNLP*, 2005, pp. 915–922.

[74] Lin Zhao, Lide Wu, and Xuanjing Huang, "Using query expansion in graph-based approach for query-focused multi-document summarization," *Information Processing and Management*, vol. 45, pp. 35–41, 2009.

[75] Andreas Stolcke, Barry Chen, Horacio Franco, Venkata Ramana Rao Gadde, Martin Graciearena, Mei-Yuh Hwang, Katrin Kirchhoff, Arindam Mandal, Nelson Morgan, Xin Lin, Ng Tim, Mari Ostendorf, Kemal Sönmez, Anand Venkataraman, Dimitra Vergyri, Wen Wang, Jing Zheng, and Qifeng Zhu, "Recent innovations in speech-to-text transcription at SRI-ICSI-UW," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, pp. 1729–1744, 2006.

[76] Adam Janin, Andreas Stolcke, Joe Frankel, Özgür Çetin, Kofi Boakye, Xavier Anguera, and Chuck Wooters, "The ICSI-SRI spring 2006 meeting speech-to-text system," in *Proceedings of MLMI*, 2006, pp. 444–456.

[77] Helmut Schmid, "Probabilistic part-of-speech tagging using decision trees," in *Proceedings of International Conference on New Methods in Language Processing*, 1994, pp. 44–49.

[78] Thorsten Brants, "TnT – A statistical part-of-speech tagger," in *Proceedings of the 6th Applied NLP Conference*, 2000, pp. 224–231.

[79] Hal Daumé III, "Notes on CG and LM-BFGS optimization of logistic regression," *http://www.cs.utah.edu/~hal/docs/daume04cg-bfgs.pdf*, 2004.

[80] Ronald S. Cheung and Bruce A. Eisenstein, "Feature selection via dynamic programming for text-independent speaker identification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, pp. 397–403, 1978.

[81] Yang Liu, Shasha Xie, and Fei Liu, "Using n-best recognition output for extractive summarization and keyword extraction in meeting speech," in *Proceedings of ICASSP*, 2010.

[82] Gabriel Murray and Giuseppe Carenini, "Summarizing spoken and written conversations," in *Proc. of EMNLP*, 2008.

[83] Ani Nenkova and Rebecca Passonneau, "Evaluating content selection in summarization: The pyramid method," in *Proc. of HLT-NAACL*, 2004.

[84] Fei Liu, Feifan Liu, and Yang Liu, "A supervised framework for keyword extraction from meeting transcripts," *IEEE Trans. on Audio, Speech, and Language Processing*, 2010.

[85] James Clarke and Mirella Lapata, "Global inference for sentence compression: An integer linear programming approach," *Journal of Artificial Intelligence Research*, vol. 31, pp. 273–381, 2008.

[86] Kevin Knight and Daniel Marcu, "Summarization beyond sentence extraction: A probabilistic approach to sentence compression," *Artificial Intelligence*, vol. 139, pp. 91–107, 2002.

[87] Trevor Cohn and Mirella Lapata, "Sentence compression as tree transduction," *Journal of Artificial Intelligence Research*, vol. 34, pp. 637–674, 2009.

[88] Jenine Turner and Eugene Charniak, "Supervised and unsupervised learning for sentence compression," in *Proc. of ACL*, 2005.

[89] Michel Galley and Kathleen McKeown, "Lexicalized markov grammars for sentence compression," in *Proc. of NAACL/HLT*, 2007.

[90] Tadashi Nomoto, "Discriminative sentence compression with conditional random fields," *Information Processing and Management*, vol. 43, pp. 1571 – 1587, 2007.

[91] Tadashi Nomoto, "A generic sentence trimmer with CRFs," in *Proc. of ACL*, 2008.

[92] Tadashi Nomoto, "A comparison of model free versus model intensive approaches to sentence compression," in *Proc. of EMNLP*, 2009.

[93] Gabriel Murray and Steve Renals, "Dialogue act compression via pitch contour preservation," in *Proceedings of ICSLP*, 2006.

[94] Eugene Charniak and Mark Johnson, "Edit detection and parsing for transcribed speech," in *Proc. of NAACL*, 2001.

[95] Yang Liu, Feifan Liu, Bin Li, and Shasha Xie, "Do disfluencies affect meeting summarization? a pilot study on the impact of disfluencies," in *Proc. of MLMI*, 2007.

[96] Fei Liu and Yang Liu, "From extractive to abstractive meeting summaries: Can it be done by sentence compression?," in *Proc. of ACL*, 2009.

[97] David M. Zajic, Jimmy Lin, Bonnie Dorr, and Richard Schwartz, "Sentence compression as a component of a multi-document summarization system," in *Proc. of DUC*, 2006.

[98] Asli Celikyilmaz, Dilek Hakkani-Tur, and Junlan Feng, "Probabilistic model-based sentiment analysis of twitter messages," in *Proceedings of the IEEE Workshop on Spoken Language Technology*, 2010, pp. 79–84.

[99] Richard Sproat, Alan W. Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards, "Normalization of non-standard words," *Computer Speech and Language*, vol. 15, no. 3, pp. 287–333, 2001.

[100] Paul Cook and Suzanne Stevenson, "An unsupervised model for text messages normalization," in *Proceedings of the NAACL HLT Workshop on Computational Approaches to Linguistic Creativity*, 2009, pp. 71–78.

[101] Monojit Choudhury, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar, and Anupam Basu, "Investigation and modeling of the structure of texting language," *International Journal on Document Analysis and Recognition*, vol. 10, no. 3, pp. 157–174, 2007.

[102] Dong Yang, Yi cheng Pan, and Sadaoki Furui, "Automatic chinese abbreviation generation using conditional random field," in *Proceedings of the NAACL HLT*, 2009, pp. 273–276.

[103] Deana L. Pennell and Yang Liu, "Normalization of text messages for text-to-speech," in *Proceedings of the ICASSP*, 2010, pp. 4842–4845.

[104] Kristina Toutanova and Robert C. Moore, "Pronunciation modeling for improved spelling correction," in *Proceedings of the ACL*, 2002, pp. 144–151.

[105] AiTi Aw, Min Zhang, Juan Xiao, and Jian Su, "A phrase-based statistical model for SMS text normalization," in *Proceedings of the COLING/ACL*, 2006, pp. 33–40.

[106] Catherine Kobus, François Yvon, and Géraldine Damnati, "Normalizing SMS: Are two metaphors better than one?," in *Proceedings of the COLING*, 2008, pp. 441–448.

[107] Richard Beaufort, Sophie Roekhaut, Louise-Amélie Cougnon, and Cédrick Fairon, "A hybrid rule/model-based finite-state framework for normalizing sms messages," in *Proceedings of the ACL*, 2010, pp. 770–779.

[108] Sasa Petrovic, Miles Osborne, and Victor Lavrenko, "The Edinburgh twitter corpus," in *Proceedings of the NAACL HLT Workshop on Computational Linguistics in a World of Social Media*, 2010, pp. 25–26.

[109] William B. Cavnar and John M. Trenkle, "N-gram-based text categorization," in *Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval*, 1994, pp. 161–175.

[110] Andrew McCallum, "MALLET: A machine learning for language toolkit," *http://mallet.cs.umass.edu/*, 2002.

[111] John Lafferty, Andrew McCallum, and Fernando Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the ICML*, 2001, pp. 282–289.

[112] Taku Kudo, "CRF++: Yet another CRF took kit," *http://crfpp.sourceforge.net/*, 2005.

[113] Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif, "Applying many-to-many alignments and hidden Markov models to letter-to-phoneme conversion," in *Proceedings of the HLT/NAACL*, 2007, pp. 372–379.

[114] Matthew Hindson, "English language hyphenation dictionary," *http://www.hindson.com.au/wordpress/2006/11/11/english-language-hyphenation-dictionary/*, 2006.

[115] Mindaugas Idzelis, "Jazzy: The java open source spell checker," *http://jazzy.sourceforge.net/*, 2005.

[116] Crispin Thurlow, "Generation txt? the sociolinguistics of young people's text-messaging," *Discourse Analysis Online*, 2003.

VITA

Fei Liu earned the B.S. degree in Computer Science and M.S. degree in Computer Science and Engineering from Fudan University, Shanghai, China, in 2004 and 2007, respectively. Since the fall of 2007, she has been a Ph.D. candidate and research assistant in the Human Language Technology Research Institute (HLTRI) at the University of Texas at Dallas. She is a recipient of the Erik Jonsson Distinguished Research Scholarship from 2007 to 2011. During the summer and fall of 2010, she worked as a research intern in Bosch Research and Technology Center in Palo Alto, CA.

Fei Liu's research interests include natural language processing, machine learning, spoken language processing, keyword extraction, generic and query-focused summarization.

Publications:

[1] Fei Liu, Feifan Liu, and Yang Liu, "A supervised framework for keyword extraction from meeting transcripts." *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 3, pp. 538–548, March 2011.

[2] Fei Liu and Yang Liu, "Using spoken utterance compression for meeting summarization: A pilot study." In *Proceedings of the 2010 IEEE Workshop on Spoken Language Technology (IEEE SLT)*, Berkeley, California, 2010.

[3] Fei Liu and Yang Liu, "Exploring speaker characteristics for meeting summarization." In *Proceedings of the* 11^{th} *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Makuhari, Japan, 2010.

[4] Yang Liu, Shasha Xie, and Fei Liu, "Using n-best recognition output for extractive summarization and keyword extraction in meeting speech." In *Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, Texas, 2010.

[5] Fei Liu and Yang Liu, "From extractive to abstractive meeting summaries: Can it be done by sentence compression?" In *Proceedings of Joint Conference of the* 47^{th} *Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, Singapore, 2009.

[6] Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu, "Unsupervised approaches to automatic keyword extraction using meeting transcripts." In *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies Conference (NAACL-HLT)*, Boulder, Colorado, 2009.

[7] Fei Liu, Feifan Liu, and Yang Liu, "Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion." In *Proceedings of the IEEE Workshop on Spoken Language Technology (IEEE SLT)*, Goa, India, 2008.

[8] Fei Liu and Yang Liu, "What are meeting summaries? An analysis of human extractive summaries in meeting corpus." In *Proceedings of the* 9^{th} *SIGDial Workshop on Discourse and Dialogue (SIGDial)*, Columbus, Ohio, 2008.

[9] Junkuo Cao, Lide Wu, Xuanjing Huang, Yaqian Zhou, and Fei Liu, "Using multiple combined ranker for answering definitional questions." In *Proceedings of the* 4^{th} *Asia Information Retrieval Symposium (AIRS)*, Harbin, China, 2008.

[10] Fei Liu, Xuanjing Huang, and Lide Wu, "Approach for extracting thematic terms based on association rules." In *Computer Engineering*, vol. 34, no. 7, pp. 81–83, 2008.