Dedicated to my family.
AUTOMATIC EXTRACTIVE SUMMARIZATION ON MEETING CORPUS

by

SHASHA XIE, 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

December, 2010
PREFACE

This dissertation (or thesis) was produced in accordance with guidelines which permit the inclusion as part of the dissertation (or thesis) the text of an original paper or papers submitted for publication. The dissertation (or thesis) 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 (or thesis) 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 (or thesis) attest to the accuracy of this statement.
ACKNOWLEDGMENTS

I would like to thank the following people that gave me great support in the last four and half years during my Ph.D. study at the University of Texas at Dallas.

First, I own my deepest appreciation to my advisor Prof. Yang Liu for her academic and moral support over the past few years. Natural language processing is totally new area for me since I changed my major from electrical engineering to computer science. Her advice and guidance are in so many aspects, from the basic knowledge of machine learning and natural language processing, selecting the research topics, to the details of my every publications. I feel really fortunate that I can finish my Ph.D. study under her guidance.

I would like to thank my committee members: Dr. John Hansen, Sanda Harabagiu, Vicent Ng and Vasileios Hatzivassiloglou for their time and effort spent on this thesis. They provided very valuable feedback and helpful suggestions to my research.

I also want to thank my labmates, Feifan Liu, Bin Li, Fei Liu, Melissa Sherman, Deana Pennell, Je Hun Jeon, Thamar Solorio, Keyur Gabani, Dong Wang, Zhonghua Qu, and Rui Xia, for their useful research discussions and great help in my life.

Part of my research was done during my visiting at International Computer Science Institute (ICSI). Many thanks to my advisors at ICSI, Dilek Hakkani-Tür and Benoit Favre, for their helpful instructions. I also want to thank my colleagues at ICSI for their help and suggestions, the leader of the speech group, Nelson Morgan, my officemates Dan Gillick, Sibel Yaman, Boriska Toth, Korbinian Rieand, and other researches at ICSI.

Finally, I want to thank Hui Lin for his greatest support on my study and life. Thank my family for their everlasting understanding and support. Your love is always in my heart.

November, 2010
With massive amounts of speech recordings available, an important problem is how to efficiently process these data to meet the user’s need. Automatic summarization is very useful techniques that can help the users browse a large amount of data. This thesis focuses on automatic extractive summarization on meeting corpus. We propose improved methods to address several issues in existing text summarization approaches, as well as leverage speech specific information for meeting summarization.

First we investigate unsupervised approaches. Two unsupervised frameworks are used in this thesis for summarization, Maximum Marginal Relevance (MMR) and the concept-based global optimization approach. We evaluate different similarity measures under the MMR framework to better measure the semantic level information. For the concept-based method, we proposed incorporating and leveraging sentence importance weights so that the extracted summary can cover both important concepts and sentences.

Second we treat extractive summarization as a binary classification problem, and adopt supervised
learning methods. In this approach, each sentence is represented by a rich set of features, and positive or negative label is assigned to indicate whether the sentence is in the summary or not. We evaluate the contribution of different features for meeting summarization using forward feature selection. To address the imbalanced data problem and human annotation disagreement, we propose using various sampling techniques and a regression model for the extractive summarization task.

Third, we focus on speech specific information for improving the meeting summarization performance. In supervised learning, we incorporate acoustic/prosodic features. Since the prosodic and textual features can be naturally split into two conditionally independent subsets, we investigate using the co-training algorithm to improve the classification accuracy by leveraging the unlabeled data information. When using the ASR output for summarization, the summarization results are often worse than using the human transcripts because of high word error rate in meeting transcripts. We introduce using rich speech recognition results, n-best hypotheses and confusion networks, to improve the summarization performance on the ASR condition. All of these proposed methods yield significant improvement over the existing approaches.
TABLE OF CONTENTS

PREFACE v

ACKNOWLEDGMENTS vi

ABSTRACT vii

LIST OF TABLES xii

LIST OF FIGURES xv

CHAPTER 1 INTRODUCTION 1

1.1 Motivation ......................................................... 1

1.2 Main Contributions ............................................. 3

CHAPTER 2 RELATED WORK 6

2.1 Extractive Speech Summarization .................................. 7

2.1.1 Unsupervised Approaches ........................................ 7

2.1.2 Supervised Approaches ........................................... 10

2.1.3 Other Approaches .................................................. 12

2.2 From Extractive to Abstractive Summarization ...................... 13

2.3 Summarization Evaluation ......................................... 14

CHAPTER 3 CORPUS AND EVALUATION MEASUREMENT 17

CHAPTER 4 UNSUPERVISED APPROACHES FOR EXTRACTIVE MEETING SUMMARIZATION 21
4.1 Using Corpus and Knowledge-based Similarity Measure in MMR ................................ 21
  4.1.1 Maximum Marginal Relevance (MMR) ......................................................... 21
  4.1.2 Similarity Measures ....................................................................................... 22
  4.1.3 Experimental Results and Discussion ......................................................... 25

4.2 Leveraging Sentence Weights in Global Optimization Framework ...................... 27
  4.2.1 Concept-based Summarization ................................................................. 28
  4.2.2 Using Sentence Importance Weight .......................................................... 29
  4.2.3 Experimental Results .................................................................................. 30

4.3 Summary .......................................................................................................... 36

CHAPTER 5 SUPERVISED APPROACH FOR EXTRACTIVE MEETING SUMMARIZATION ...... 38

5.1 Features in Supervised Summarization Approaches .......................................... 38
  5.1.1 Lexical Features ......................................................................................... 39
  5.1.2 Structural and Discourse Features ............................................................. 40
  5.1.3 Topic-Related Features ............................................................................. 41

5.2 Improving Supervised Summarization Approaches ............................................ 42
  5.2.1 Issues in Supervised Learning for Meeting Summarization ....................... 42
  5.2.2 Addressing the Imbalanced Data Problem ................................................ 44
  5.2.3 Using Regression Model for Summarization .............................................. 50
  5.2.4 Forward Feature Selection ....................................................................... 52

5.3 Experimental Results and Discussion .................................................................. 53
  5.3.1 Baseline Results ....................................................................................... 53
  5.3.2 Results Using Sampling Methods .............................................................. 54
  5.3.3 Regression Results ..................................................................................... 61
  5.3.4 Feature Selection Results .......................................................................... 63
### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>ROUGE-1 F-measure results (%) using different similarity approaches on test set using human transcripts.</td>
<td>26</td>
</tr>
<tr>
<td>4.2</td>
<td>ROUGE-1 F-measure results (%) using different similarity approaches on test set using ASR output.</td>
<td>27</td>
</tr>
<tr>
<td>4.3</td>
<td>ROUGE-1 F-measure results (%) of three baselines on the dev set for both human transcripts (REF) and ASR output.</td>
<td>31</td>
</tr>
<tr>
<td>4.4</td>
<td>ROUGE-1 F-measure results (%) on the dev set for both human transcripts (REF) and ASR output.</td>
<td>34</td>
</tr>
<tr>
<td>4.5</td>
<td>ROUGE-1 F-measure results (%) of incorporating sentence importance weights on the dev set using both human transcripts (REF) and ASR output.</td>
<td>34</td>
</tr>
<tr>
<td>4.6</td>
<td>ROUGE-1 F-measure results (%) for different word compression ratios on test set for both human transcripts (REF) and ASR output.</td>
<td>36</td>
</tr>
<tr>
<td>5.1</td>
<td>List of lexical features.</td>
<td>39</td>
</tr>
<tr>
<td>5.2</td>
<td>List of structural and discourse features.</td>
<td>40</td>
</tr>
<tr>
<td>5.3</td>
<td>List of topic related features.</td>
<td>41</td>
</tr>
<tr>
<td>5.4</td>
<td>Example of two similar sentences with different labels.</td>
<td>43</td>
</tr>
<tr>
<td>5.5</td>
<td>Weighting measures used in sampling for non-summary sentences.</td>
<td>46</td>
</tr>
<tr>
<td>5.6</td>
<td>Example of new labels using regression, up-sampling, and down-sampling for a non-summary sentence (second row) that is similar to a summary sentence (first row).</td>
<td>51</td>
</tr>
<tr>
<td>5.7</td>
<td>ROUGE-1 F-measure results (%) of the baselines using human transcripts.</td>
<td>53</td>
</tr>
<tr>
<td>5.8</td>
<td>ROUGE-1 F-measure results (%) of the baselines using ASR outputs.</td>
<td>54</td>
</tr>
</tbody>
</table>
5.9 ROUGE-1 F-measure results (%) of up-sampling by replicating the positive samples on both human transcripts and ASR outputs. .................................................. 57

5.10 ROUGE-1 F-measure Results (%) of down-sampling by randomly removing negative samples on both human transcripts and ASR output. ........................................ 60

5.11 ROUGE-1 F-measure results (%) of combining sampling and regression. ‘Ds’ is down-sampling; ‘Rs’ is re-sampling; ‘Rg’ is regression. ........................................ 63

5.12 ROUGE-1 F-measure results (%) after Forward Feature Selection for human transcripts. ................................................................. 64

5.13 ROUGE-1 F-measure results (%) after Forward Feature Selection for ASR outputs. 65

5.14 ROUGE-1 F-measure results (%) using selected features and sampling and regression on human transcripts. ................................................................. 66

5.15 ROUGE-1 F-measure results (%) using selected features and sampling and regression on ASR output. ................................................................. 67

5.16 ROUGE-1 F-measure results (%) on test set for human transcripts. ......................... 67

5.17 ROUGE-1 F-measure results (%) on test set for ASR outputs. ................................. 68

6.1 ROUGE-1 F-measure results (%) of the baselines by selecting longest sentences and using non-prosodic features on development set. ......................... 74

6.2 ROUGE-1 F-measure results (%) using raw values and different normalization methods of acoustic/prosodic features on development set for human transcripts. .... 75

6.3 ROUGE-1 F-measure results (%) using raw values and different normalization methods of acoustic/prosodic features on development set for ASR output. ........ 75

6.4 ROUGE-1 F-measure results (%) of adding delta features on development set. ........ 77

6.5 The five most and least effective prosodic features evaluated using human transcripts on development set. ................................................................. 77

6.6 ROUGE-1 F-measure results (%) of integrating prosodic and non-prosodic information, in comparison with using only one information source on development set. ................................................................. 79
6.7 ROUGE-1 F-measure results (%) on test set. ........................................ 80
6.8 An example of 5-best hypotheses for a sentence segment. ......................... 89
6.9 ROUGE F-measureresults (%) using 1-best hypotheses and human transcripts on the development set. The summarization units are the ASR segments. ............... 96
6.10 ROUGE F-measure results (%) on the development set using different vector representations based on confusion networks: non-pruned and pruned, using posterior probabilities (“wp”) and without using them. ........................................ 104
6.11 ROUGE F-measure results (%) on the development set using different segment representations, with the summaries constructed using the corresponding human transcripts for the selected segments. ............................... 105
6.12 WER (%) of extracted summaries using n-best hypotheses and confusion networks on the development set. ............................................................. 107
6.13 ROUGE F-measure results (%) on the test set using different inputs for summarization, and different forms of summaries (using recognition hypotheses or human transcripts of the selected summary segments). ............... 111
LIST OF FIGURES

3.1 An excerpt of meeting transcripts with summary information. 18
3.2 ASR output for the excerpt shown in Figure 3.1. 19

4.1 ROUGE-1 F-measure results (%) using different percentage of important sentences during concept extraction on the dev set for both human transcripts (REF) and ASR output. The horizontal dashed lines represent the scores of the baselines using all the sentences. 32
4.2 ROUGE-1 F-measure results (%) using pruning to preserve different percentage of candidate summary sentences on the dev set for both human transcripts (REF) and ASR output. The horizontal dashed lines represent the scores of the baselines using all the sentences. 33

5.1 Illustration of up-sampling for binary classification. 45
5.2 Illustration of down-sampling in binary classification. 48
5.3 Flowchart for re-sampling method. 49
5.4 Coverage of the original positive samples (left Y-axis) and the percentage of positive samples in the selected data (right Y-axis) using TFIDF and cosine scores as selection criteria for different re-sampling rates. 50
5.5 ROUGE-1 F-measure results (%) using up-sampling on human transcripts. 56
5.6 ROUGE-1 F-measure results (%) using up-sampling on ASR outputs. 57
5.7 ROUGE-1 F-measure results (%) using down-sampling on human transcripts. 59
5.8 ROUGE-1 F-measure results (%) using down-sampling on ASR outputs. 59
5.9 ROUGE-1 F-measure results (%) using the re-sampling method. 61
5.10 ROUGE-1 F-measure results (%) using the regression model. 62
6.1 ROUGE-1 F-measure results of co-training on the development using the models trained from textual and prosodic features. ................................. 85
6.2 ROUGE-1 F-measure results of co-training results on the test using the models trained from textual (upper) and prosodic features (lower). ................. 87
6.3 An example of confusion networks for a sentence segment. ..................... 90
6.4 ROUGE-1 F-measure results using n-best lists (up to 50 hypotheses) on the development set. ................................................................. 97
6.5 ROUGE-2 F-measure results using n-best lists (up to 50 hypotheses) on the development set. ................................................................. 98
6.6 ROUGE-1 F-measure results using n-best lists (up to 50 hypotheses) on the development set, with the summaries constructed using the corresponding human transcripts of the selected hypotheses. .............................. 99
6.7 ROUGE-2 F-measure results using n-best list (up to 50 hypotheses) on the development set, with the summaries constructed using the corresponding human transcripts of the selected hypotheses. .............................. 100
6.8 Average node density and word coverage of the confusion networks on the development set. ................................................................. 102
CHAPTER 1
INTRODUCTION

1.1 Motivation

With massive amounts of speech recordings and multimedia data available, an important problem is how to efficiently process these data to meet user’s information need. There has been increasing interest recently in automatically processing the speech data, including summarization, and other understanding tasks in the research community (for example, programs such as AMI/AMIDA, CHIL in Europe [1, 2], and CALO [3], and NIST’s Rich Transcription evaluation on the meeting domain [4]).

Automatic summarization is a very useful technique to facilitate users to browse large amount of data efficiently. Summarization can be divided into different categories along several different dimensions [5]. Based on whether or not there is an input query, the generated summary can be query-oriented or generic; based on the number of input documents, summarization can use a single document or multiple documents; in terms of how sentences in the summary are formed, summarization can be conducted using either extraction or abstraction — the former only selects sentences from the original documents, whereas the latter involves natural language generation. Overall, automatic summarization systems aim to generate a good summary, which is expected to be concise, informative, and relevant to the original input.

A lot of techniques and approaches have been proposed for automatic text summarization the
past decades, and there are some benchmark tests such as TIDES, AQUAINT, MUC, DUC, and TAC [6, 7, 8, 9, 10]. To summarize and process the speech data, a natural solution is to transcribe the speech recordings to texts, and apply some well studied text summarization approaches. However, usually when the traditional NLP methods are directly applied to speech transcripts, the performance is not as good as for text processing. There are several issues that make the speech transcripts different from the written texts.

- The speech transcripts often have a lot of disfluencies, while written texts are normally well formed and organized. This is especially the case for spontaneous speech domains, such as multiparty meetings, conversational telephone speech. Even for broadcast news, which are read speech, they contain unavoidable disfluencies. In addition, there are often inserted broadcast conversations which are more conversational.

- If we use the output of the automatic speech recognition (ASR) system for summarization, the output is actually only word sequence, and there are no punctuation marks or sentence segments associated with it. Therefore we first need to segment the word sequence into pieces to use as summarization units. A simple way to do that is using acoustic segment boundaries that correspond to long stretches of silence or a change of conversational turn. There are other automatic methods to detect the sentence boundaries, which integrate more linguistic information, such as language model score, part-of-speech tags, and etc. However, these automatically segmented sentences are still different from the human annotated linguistic sentences. This often affects the down-streaming language understanding performance.

- The speech transcripts contain word errors, especially for the conversational speech. For
example, for meeting recordings, the word error rate could be as high as around 40%. With such lower accuracy, it is hard to read the transcripts and generate a summary even for human annotators. If the cue words or cue phrases are not correctly recognized, it will have great impact on the selection of important sentences.

All of these issues make speech summarization different from text summarization, and the traditional text summarization approaches generally do not perform well in speech summarization task.

This thesis focuses on the task of extractive summarization using the meeting corpus. Given the meeting document (a meeting recording together with its transcript), our aim is to select the most important and representative parts, and concatenate them together to form a summary. Automatic summarization on the meeting domain is more challenging comparing to summarization of other speech genres, such as broadcast news, lectures, and voice mail. Different from broadcast news and lectures, which are read or pre-planed speech, meetings are more spontaneous, so the meeting transcripts contain a lot of disfluencies, such as filled pauses, repetitions, revisions, and etc. Although some meetings have pre-defined topics, in general the content of meetings is less coherent than other speech genres. For speech recognition, the ASR performance in meeting domain is also worse than other speech domains. Such noisy data has great impact on the performance of traditional summarization approaches.

1.2 Main Contributions

In this thesis, we exploited different methods for the task of extractive meeting summarization, including unsupervised, supervised, and semi-supervised approaches. We proposed improved
methods to address several issues in existing summarization approaches, as well as leverage speech specific information for meeting summarization.

Two unsupervised frameworks were introduced and investigated in Chapter 4, maximum marginal relevance (MMR) and the concept-based global optimization framework. Under the MMR framework, other than the simple lexical matching, we evaluated different similarity measures to better capture the semantic relationship between text segments. Better similarity measures can help define the importance of each sentence and its relationship with other sentences in the document. The concept-based optimization framework selects a subset of summary sentences to maximize the coverage of important concepts. We proposed to leverage the sentence level information in this framework to improve the linguistic quality of extracted summaries.

In Chapter 5, supervised learning approaches were adopted, where the summarization task is considered as a binary classification problem, summary or non-summary, and each sentence is represented by a large set of features. We analyzed the feature effectiveness for both human and ASR conditions. In addition, two important issues associated with this classification approach were addressed. First, since the summary sentences are only a small portion of the entire document, there is an imbalanced data problem during classifier training. The second problem is that the agreement between different human annotators is very low. We proposed different sampling methods to solve these problems by providing more balanced training data through changing the original labels of the samples, or selecting a subset of balanced samples. We also proposed using a regression model for this task, where the label for each sample is not binary any more, but a numerical value representing its importance.

Chapter 6 addressed several issues specific to speech summarization. Since the speech record-
ings are available, in Section 6.1, we investigated whether we can extract more features from speech data to help improve the summarization performance. These features include pitch, energy, sentence duration, speaking rate, and their different normalized variants. We showed that using only the prosodic information we can generate a better summarizer than using the textual information. The textual and acoustic features can be naturally split into two conditional independent sets, and each of them is sufficient to generate a good summarizer. In Section 6.2, we adopted the co-training algorithm so that we can obtain good summarization performance by only using a small amount of labeled data and extra unlabeled data for training.

Because of the high word error rates in meeting transcripts, we found that using the ASR output often degrades the summarization performance comparing to using the human annotated transcripts. In Section 6.3, we demonstrated the feasibility of using rich speech recognition results to improve the speech summarization performance. Two kinds of structures were considered, n-best hypotheses and confusion networks. Under an unsupervised MMR framework, we proposed the term weighting and vector representation methods to reframe the text segments considering more word candidates and sentence hypotheses.
CHAPTER 2
RELATED WORK

In this chapter, we provide a literature review of previous research related to automatic speech summarization. We limit our survey to speech summarization, since the focus of this thesis is extractive meeting summarization.

Automatic speech summarization has received a lot of attentions in recent years, and different speech domains have been explored. Broadcast news speech is the first domain to be exploited for speech summarization [11]. This domain is similar to the news article domain widely used for text summarization, but is different as it consists of read and spontaneous speech, and there are speech recognition errors (though generally lower than other speech genres), thus presents a good starting point to evaluate the portability of the classical features and approaches used in text summarization. Research was then expanded to lecture speech summarization [12]. Lectures are much longer than broadcast news, and contain more spontaneous speech. However, the slides associated with the lectures provide additional information for selecting the summary sentences. Comparing to the previous two speech genres, meeting speech is the most spontaneous one, and often involves multiple speakers [13]. Recently there have been many efforts on various meeting understanding tasks, such as automatic summarization, meeting browsing, detection of decision parts and action items, topic segmentation, keyword extraction, and dialog act tagging [13, 14, 15, 16, 17, 18]. Other speech genres used for summarization include television shows, conversational telephone speech, and voice mail [19, 20, 21].
In the following sections, we first introduce the state-of-the-art approaches used for automatic extractive speech summarization. Then we briefly describe the recent activities of generating abstractive summaries. Last, we discuss research on evaluation metrics for summarization performance.

2.1 Extractive Speech Summarization

2.1.1 Unsupervised Approaches

Unsupervised approaches are relatively simple, and robust to different corpora. The summary sentences are usually selected according to their own importance, and their relationship to other sentences in the document. In [22], the authors proposed to select important utterances or n-grams using each word’s inverse document frequency and acoustic confidence score, and found that this can help the summarizer select more accurate utterances. In [23], the summary sentences were extracted according to a sentence’s significance score measured using TFIDF, trigram probability of the sentence, and confidence score from the ASR system.

In the above two approaches, the summary sentences were selected based only on their individual significance. Other than the sentence importance, the relationship between sentences should also be considered during the sentence selection, since the generated summary should be concise and representative, and no redundant information should be included. MMR was introduced in [24] for text summarization, and has been applied to speech summarization. This algorithm can select the most relevant sentences, and at the same time avoid the redundancy by removing the sentences that are too similar to the already selected ones. The summary sentences are selected
iteratively, and the weight for each sentence is calculated using the following equation:

\[
MMR(S_i) = \lambda \times Sim_1(S_i, D) - (1 - \lambda) \times Sim_2(S_i, Summ) \tag{2.1}
\]

where \( D \) is the document vector, \( Summ \) represents the sentences that have been extracted into the summary, and \( \lambda \) is used to adjust the combined score to emphasize the relevance or to avoid redundancy. In [25], the authors compared the MMR method with two other approaches: one selecting the sentences similar to the first sentence, and the other selecting sentences that are most similar to the entire document. They showed that the MMR method outperformed the other two on spontaneous speech summarization.

Latent semantic analysis (LSA) approach has been used in extracting the summary sentences by exploring the semantic similarity between sentences considering a set of latent topics [13]. In this method, a set of latent topics \( T_1, T_2, \ldots, T_K \) is first defined, and then each word in the document is modeled to be generated from these latent topics.

\[
P(w_i|D) = \sum_{k=1}^{K} P(w_i|T_k)P(T_k|D) \tag{2.2}
\]

In [26], the authors proposed two methods under LSA, topic significance and term entropy, where the important terms were detected and given higher weights during calculation. The term importance was calculated according to the term’s distribution in a large corpus, or term entropy over the latent topics. They found that their proposed methods obtained better summarization performance comparing to the original LSA method.

In [27], the authors introduced a concept-based global optimization framework, where concepts were used as the minimum units, and the important sentences were extracted to cover as many
concepts as possible. A global optimization function was defined as following:

\[
\begin{align*}
\text{maximize} & \quad \sum_i w_i c_i \\
\text{subject to} & \quad \sum_j l_j s_j < L
\end{align*}
\]

where \(w_i\) is the weight of concept \(i\), \(c_i\) is a binary variable indicating the presence of that concept in the summary, \(l_j\) is the length of sentence \(j\), \(L\) is the desired summary length, and \(s_j\) represents whether a sentence is selected for inclusion in the summary. Integer linear programming method was used to select sentences that maximize the objective function under the length constraint \(L\). The authors showed that this global optimization method outperformed MMR.

Graph-based methods, such as LexRank [28], represent a document using a graph, where sentences are modeled as nodes. Then the summary sentences are ranked according to their similarities with other nodes. [29] proposed ClusterRank, a modified graph-based method, to cope with high noise and redundancy in spontaneous speech transcripts. In ClusterRank, the neighbor sentences were first clustered according to their cosine similarity scores, and the graph was constructed on these clusters. The similarity between clusters was calculated by only considering the important words contained in the cluster. The authors showed better performance of using ClusterRank than LexRank. In [30], the authors suggested formulating the summarization task as optimizing submodular functions defined on the document’s semantic graph. The construction of the graph was similar to LexRank, where sentences were modeled as nodes, and similarities between nodes as edge weights. The optimization was theoretically guaranteed to be near-optimal under the framework of submodularity. The authors achieved significantly better results than using MMR and the concept-based global optimization framework.
2.1.2 Supervised Approaches

Another line of work for extractive speech summarization is based on supervised methods, where all the utterances in a document are divided into two classes, in summary or not, then the summarization task can be considered as a binary classification problem. Although a large amount of labeled data is necessary for training the classifier, supervised approaches usually achieve better performance comparing to the unsupervised ones. Various models have been investigated for this classification task, such as Bayesian network [31], maximum entropy [32], support vector machines (SVM) [20], hidden Markov model (HMM) [33], and conditional random fields (CRF) [34]. In [13], Murray et al. compared MMR, LSA, and the feature-based classification approach, and showed that human judges favor the feature-based approaches.

In [31], the authors constructed a summarizer using the structural features. Different from text summarization, other than sentence position and sentence length, the authors included the speaker-related features, which represent the overall contribution of each speaker. The authors reported that the system was robust to the speech recognition errors by comparing the results obtained from using manual transcripts and ASR outputs. In [35], the authors provided an empirical study of the usefulness of different types of features in the domain of broadcast news summarization, including lexical, acoustic/prosodic, structural, and discourse features. The lexical and discourse features are also commonly used in text summarization, such as the count of name entities, the number of words in the sentence, and the features representing the word distributions. The structural features were similar as introduced in [31], which include the speaker information. The acoustic/prosodic features are unique for speech summarization, which are extracted from the speech recordings. The authors pointed out that a change in pitch, amplitude or speaking rate may signal differences in the
relative importance of the speech segments. Experimental results showed that a combination of all the features performs the best. However, acoustic/prosodic and structural features were enough to build a “good” summarizer when speech transcripts are not available.

In [36], the authors compared the contribution of different types of features in conversational speech summarization using the Switchboard data. Other than the features mentioned before, they included MMR scores of each sentence, and a set of spoken-language features. These spoken-language features contained the number of repetitions and filled pauses, which can better capture the characteristics of more spontaneous speech. Their experiments showed that speech disfluencies helped identify important utterances, while the structural features are less effective than in broadcast news.

In [37], the authors introduced the rhetorical information for lecture summarization. Since lectures and presentations are planned speech and follow a relatively rigid rhetorical structure, the authors proposed to use a hidden Markov model to learn this rhetorical structure trained on the slides associated with the speech, and features were then extracted for each structure. They proved that using rhetorical structure improved the summarization performance. They also showed that lexical features were more important than acoustic features, and different from broadcast news and conversational speech summarization, discourse features were not useful for lecture summarization.

On the task of meeting summarization, Murray et al. [38] analyzed the speaker activity, turn-taking, and discourse cues, and reported that using these features was advantageous and efficient than only using the textual features.

The contribution of different types of features varies according to different speech genres. In
[12], the authors compared the lexical, structural, and acoustic features on speech summarization of Mandarin broadcast news and lecture speech. They found that structural features were superior to acoustic and lexical features when summarizing broadcast news, but acoustic and structural features made more important contribution to broadcast news summarization comparing to lecture summarization.

2.1.3 Other Approaches

Other than unsupervised and supervised approaches, previous research investigated the ways that can combine the unsupervised and supervised methods. The results from unsupervised methods can be considered as one of the features for supervised learning. For example, in [39], the MMR results were used as features for supervised training. In [40], the authors formulated extractive summarization as a risk minimization problem and proposed a unified probabilistic framework that naturally combined the supervised and unsupervised approaches. The summary sentences were selected using the following function:

$$S = \arg\min_{S_i \in D} \sum_{S_j \in D} L(S_i, S_j) \frac{P(D|S_j)P(S_j)}{\sum_{S_m \in D} P(D|S_m)P(S_m)}$$

(2.5)

where $D$ is the document to be summarized, and $S_i$ is one of the sentences contained in $D$. $P(D|S_j)$ can be modeled as a sentence generative model, where the probability is calculated using the product of each word’s generative model $P(w|S_j)$. The sentence prior probability $P(S_j)$ was in proportion to the posterior probability of the sentence being included in the summary class obtained from a supervised learning classifier. The concept of MMR was incorporated into the computation of loss function $L(S_i, S_j)$. Such a framework can be regarded as a generalization of several existing summarization methods. The authors showed significant improvements over several popular summarization methods, such as vector space model, LexRank, and CRF.
2.2 From Extractive to Abstractive Summarization

The extracted summary sentences often contain a lot of disfluencies, and may have word errors if the input for summarization is automatically recognized transcripts. Simply concatenating extracted sentences may not comprise a good summary. Other than extractive summarization, researchers also worked on generating abstractive summaries, or compressing the extracted sentences and merging them into a more concise summary.

Instead of post-processing the extractive summary sentences, some previous research removed the disfluencies in the documents before performing summarization [41]. The types of disfluencies were defined as filled pauses, restarts or repairs, and false starts. A part-of-speech tagger was trained to detect the filled pauses, and the words tagged with CO (coordinating), DM (discourse marker), and ET (editing term), were removed from the texts. The false starts were detected using a decision tree, and a rule-based script was used to detect the repetition. After pre-processing the speech transcripts, MMR was used to select the important summary sentences.

In [42], for each sentence, a set of words maximizing a summarization score was extracted from automatically transcribed speech. This extraction was performed using a dynamic programming technique. The summarization score consisted of a word significance measure, a confidence measure, linguistic likelihood, and a word concatenation probability which was determined by a dependency structure in the original speech. These word sequences were then concatenated to form a summary. The disfluencies and possible errorful words will be automatically removed from the summary because the word sequence was selected considering acoustic confidence scores and linguistic likelihood.

In [43], the abstractive summaries were generated by applying sentence compression on se-
lected extractive summary sentences. The authors proposed several compression methods. First the filler phrases were detected and removed, 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 meaning, such as, “for example”, “of course”, and “sort of”. Then an integer programming based framework was introduced, where the word sequence was selected maximizing the sum of the significance scores of the consisting words and n-gram probabilities from a language model. The experimental results showed that further compression of the extractive summaries can improve the summarization performance.

In [44], the authors developed an abstractive conversation summarization system consisting of interpretation and transformation components. In the interpretation component, each sentence is mapped to a simple conversation ontology, where conversation participants and entities are linked by object properties, such as decisions, actions, and subjective opinions. In the transformation step, a summary is created by maximizing a function relating sentence weights and entity weights. The authors pointed out that these selected summary sentences corresponded to \(< participant, relation, entity >\) triplets in the ontology, for which they can subsequently generate novel text by creating linguistic annotations of the conversation ontology. This result can also be used to generate structured extracts by grouping sentences according to specific phenomena such as action items and decisions.

2.3 Summarization Evaluation

How to properly evaluate summarization results automatically is still an open topic. Summarization evaluation techniques can generally be classified as intrinsic or extrinsic evaluation.
Intrinsic evaluation compares the system generated summaries with gold-standard human summaries. Extrinsic metrics, on the other hand, evaluate the usefulness of the summary in performing a real-world task. Most of the summarization work evaluates their performance using intrinsic measures, because such evaluations are easily replicable and more useful for development purposes.

ROUGE [45] has been widely used in previous research and benchmark summarization tests (e.g., DUC). ROUGE compares the system generated summary with reference summaries (there can be more than one reference summary), and measures different matches, such as N-gram, longest common sequence, and skip bigrams. For example, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries computed as follows:

\[
ROUGE - N = \frac{\sum_{S \in \text{ReferenceSummaries}} \sum_{\text{gram}_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \text{ReferenceSummaries}} \sum_{\text{gram}_n \in S} \text{Count}(\text{gram}_n)}
\]

However, some research showed that in general the correlation of ROUGE scores and human evaluation is low for some speech domains [46, 47].

In [48], the authors proposed a pyramid method for summarization evaluation based on Summarization Content Units (SCU). The annotation starts with identifying similar sentences, and then SCUs are extracted by inspecting and identifying more tightly related subparts. Each SCU has a weight corresponding to the number of summaries it appears in. After the annotation procedure is completed, a pyramid is built based on the weight of each SCU. The score for the automatically generated summary is assigned as a ratio of the sum of the weights of its SCUs to the sum of the weights of an optimal summary with the same number of SCUs. This pyramid method not only assigns a score to a summary, but also allows the investigator to find what important information is missing, and thus can be directly used to target improvements of the summarizer. However,
creating an initial pyramid is very laborious. In [34], the authors adopted this Pyramid evaluation metric but with the constraints that two summary units are considered equivalent if and only if they are extracted from the same location in the original document.

In [38], the authors proposed to evaluate the summarization performance using weighted precision and recall at the dialog act level, which utilized the particular summary annotation of the corpus they used in their study. The evaluation was conducted according to how often each annotator linked a given extracted dialog act to a summary sentence. The weighted precision, recall and f-score were calculated as the final score for the summary to be evaluated.
CHAPTER 3
CORPUS AND EVALUATION MEASUREMENT

The data we use in this thesis is the ICSI meeting corpus [49]. It contains 75 recordings from natural meetings (most are research discussions in speech, AI, and networking areas) [49]. Each meeting is about an hour long and has multiple speakers. These meetings have been transcribed, annotated with dialog acts (DA) [50], topic segmentation, and extractive summaries [46]. For extractive summary annotation, the annotators were asked to select and link DAs from the transcripts that are related to each of the sentences in the provided abstractive summaries (see [46] for more information on annotation). Figure 3.1 shows a sample from one of the human transcripts, where each line corresponds to a DA, and the ID at the beginning of each line (marked by S*) is the speaker ID. In this excerpt, three sentences (18, 19, and 25) were marked as the summary sentences by the annotator. From this example, we can see the meeting transcripts are significantly different from the input for text summarization (e.g., news article) in that it is very spontaneous, contains disfluencies and incomplete sentences, has low information density, and involves multiple speakers.

The automatic speech recognition (ASR) output for this corpus is obtained from a state-of-the-art SRI conversational telephone speech system [51, 52]. The word error rate is about 38.2% on the entire corpus. We align the human transcripts and ASR output, then map the human annotated DA boundaries and topic boundaries to the ASR words, such that we have human annotation for the ASR output. For the extractive summarization task, we use human annotated DA boundaries
S1 yeah if you breathe under breathe and then you see af go off then you know -pau- it's p- picking up your mouth noise [laugh]
[S2 oh that's good
[S3 cuz we have a lot of breath noises
[S3 yep
[S3 test [laugh]
[S2 in fact if you listen to just the channels of people not talking it's like [laugh]
[S2 it's very disgust-
[S3 what
[S3 did you see hannibal recently or something
[S2 sorry
[S2 exactly
[S2 it's very disconcerting
[S2 ok
[S2 so um
[S2 i was gonna try to get out of here like in half an hour
[S2 um
[S2 cuz i really appreciate people coming

* [18] S2 and the main thing that i was gonna ask people to help with today is -pau- to give input on what kinds of database format we should -pau- use in starting to link up things like word transcripts and annotations of word transcripts
*S[19] S2 so anything that transcribers or discourse coders or whatever put in the signal with time-
marks for like words and phone boundaries and all the stuff we get out of the forced alignments and
the recognizer
[S2 so we have this um
[S2 i think a starting point is clearly the the channelized -pau- output of dave gelbart's program
[S2 which don brought a copy of
[S3 yeah
[S3 yeah i'm i'm familiar with that
*S[25] S3 i mean we i sort of already have developed an xml format for this sort of stuff
[S2 um
[S2 which
[S1 can i see it
[S3 and so the only question is it the sort of thing that you want to use or not
[S3 have you looked at that
[S3 i mean i had a web page up
[S2 right

Figure 3.1. An excerpt of meeting transcripts with summary information.
as sentence information and perform sentence-based extraction. The ASR output for the same example as in Figure 3.1 is shown in Figure 3.2. We can see a lot of recognition errors in this example.

```
1. yeah if you breathe under breathing in the caf boston you know picking up your mouth right
2. well that's good
3. because we have a lot of breath
4. yep
5. test
6. in fact if you listen to to just the channels of people not talking to
7. what
8. did you see a cannibal recent years something
9. it's very disconcerting
10. okay
11. so um
12. i was going to try to get out of here like in half an hour
13. um
14. because i really appreciate people coming
15. *[18] and the main thing that i was going to ask people to help with today is two give input on what kinds of database format we should use and starting to link up things like word transcription indications of we're transcript
16. *[19] so anything that transcribers or discourse toters whatever put in the say no with tire marks for like words and phone boundaries and all the stuff we get out of the forced alignment and the recognize or
17. so we have this um
18. i think a starting point is clearly that the generalized output of dave cal arts program
19. which don brought a copy of
20. yeah
21. i'm i'm familiar with
22. *[25] i mean we i sort of already have developed an x. m. l. format for this sort of stuff
23. um
24. which
25. and so the only question is that the sort of thing that you want to use or not
26. have you looked at that
27. mean i had a web page up
28. right
```

Figure 3.2. ASR output for the excerpt shown in Figure 3.1.

The same 6 meetings as in previous work (e.g., [13, 27, 30, 53]) are used as the test set in this study. Furthermore, 6 other meetings were randomly selected from the remaining 69 meetings in the corpus to form a development set, then the rest is used to compose the training set for the supervised learning approach. Each of the meetings in the training and development set has only one human-annotated summary, whereas for the test meetings, we use 3 reference summaries
from different annotators for evaluation. For summary annotation, human agreement is quite low [54]. The average Kappa coefficients among these 3 annotators on the test set ranges from 0.211 to 0.345. The lengths of the reference summaries are not fixed and vary across annotators and meetings. The average word compression ratio for the test set is 14.3%, and the mean deviation is 2.9%. These statistics are similar for the training set.

In our study we use ROUGE as the evaluation metrics because it has been used in previous studies of speech summarization, and thus we can compare our work with previous results [13, 20, 27, 34, 35, 37, 40]. The options we used in this study are the same as those used in DUC: stemming summaries using Porter stemmer before computing various statistics (-m); averaging over the sentence unit ROUGE scores (-t 0); assigning equal importance to precision and recall (-p 0.5); computing statistics in the confidence level of 95% (-c 95) based on sampling points of 1000 in bootstrap resampling (-r 1000).
CHAPTER 4
UNSUPERVISED APPROACHES FOR EXTRACTIVE MEETING SUMMARIZATION

In this chapter, we study two unsupervised approaches for extractive meeting summarization, maximum marginal relevance (MMR) and a concept-based global optimization framework. Among all the approaches for summarization, MMR is one of the simplest techniques, and has been effectively used for text and speech summarization [24]. The extractive summarization problem can also be modeled using a global optimization framework based on the assumption that sentences contain independent concepts of information, and that the quality of a summary can be measured by the total value of unique concepts it contains [27]. In this chapter, we propose improved solutions for these two unsupervised methods to obtain better summarization performance.

4.1 Using Corpus and Knowledge-based Similarity Measure in MMR

4.1.1 Maximum Marginal Relevance (MMR)

MMR is a greedy algorithm, where the summary sentences with the highest scores are selected iteratively as we introduced in Section 2.1.1. The score is calculated using two similarity functions ($Sim_1$ and $Sim_2$), as shown in Equation 2.1, representing the similarity of a sentence to the entire document and to the selected summary, respectively. We adopt two approximated methods to speed up the process of calculating the MMR scores [55]. For each sentence, we calculate its similarity to all the other sentences that have a higher similarity score to the document (according to the results
of $Sim_1$), and use it as an approximation for $Sim_2$. Therefore, the summary selection process only needs to find the top sentences that have high combined scores, which is an offline processing. Another approximation we use is not to consider all the sentences in the document, but rather only a small percent of sentences (based on a predefined percentage) that have a high similarity score to the entire document. Our hypothesis is that the sentences that are closely related to the document are worth being selected.

4.1.2 Similarity Measures

An important part in MMR is how we can appropriately represent the similarity of two text segments. In this section, we evaluate three different similarity measures.

**Cosine Similarity**

One commonly used similarity measure is cosine similarity, which we use as our baseline in this study. In this approach, each document (or a sentence) is represented using a vector space model. The cosine similarity between two vectors ($D_1, D_2$) is:

$$sim(D_1, D_2) = \frac{\sum_i t_{1i}t_{2i}}{\sqrt{\sum_i t_{1i}^2} \times \sqrt{\sum_i t_{2i}^2}}$$

(4.1)

where $t_i$ is the term weight for a word $w_i$, for which we use the TF-IDF (term frequency, inverse document frequency) value, as widely used in information retrieval. The IDF weighting is used to represent the specificity of a word: a higher weight means a word is specific to a document, and a lower weight means a word is common across many documents. IDF values are generally obtained from a large corpus as follows:

$$IDF(w_i) = log(N/N_i)$$

(4.2)
where $N_i$ is the number of documents containing $w_i$ in a collection of $N$ documents. In [56], Murray and Renals compared different term weighting approaches to rank the importance of the sentences (simply based on the sum of all the term weights in a sentence) for meeting summarization, and showed that TF-IDF weighting is competitive.

**Centroid Score**

Another distance measure we evaluate is the centroid score [55], which only considers the salient words for the similarity between a sentence and the entire document. The same vector representation is used as in cosine similarity. In this approach, each word in a sentence $S_i$ is checked to see if it occurs in the text segment $T$ and if the term weight (TF-IDF value) of this word is greater than a predefined threshold. If these requirements are met, the term weight of this word is added to the centroid score for the sentence.

$$Score_{centroid}(i) = \sum_{w_j \in S_i} \text{bool}(w_j \in T) \times \text{bool}(tw(w_j) > v) \times tw(w_j)$$ (4.3)

where $tw(w_j)$ represents the term weight for the word $w_j$, and the functions $\text{bool}(w_j \in T)$ and $\text{bool}(tw(w_j) > v)$ check the two conditions mentioned above. In the MMR system, we use the centroid score as the first similarity function ($Sim_1$ in Equation 2.1). The second similarity measure $Sim_2$ is still the cosine distance.

**Corpus-based Semantic Similarity**

The cosine and centroid scores between a sentence and a document are all based on simple lexical matching, that is, only the words that occur in both contribute to the similarity. Such literal
comparison can not always capture the semantic similarity of text. Therefore we use the following function to compute the similarity score between two text segments [57].

\[
sim(T_1, T_2) = \frac{1}{2} \left( \frac{\sum_{w \in \{T_1\}} \text{maxSim}(w, T_2) \cdot \text{idf}(w)}{\sum_{w \in \{T_1\}} \text{idf}(w)} + \frac{\sum_{w \in \{T_2\}} \text{maxSim}(w, T_1) \cdot \text{idf}(w)}{\sum_{w \in \{T_2\}} \text{idf}(w)} \right) \quad (4.4)
\]

\[
\text{maxSim}(w, T_i) = \max_{w_i \in \{T_i\}} \{ \text{sim}(w, w_i) \} \quad (4.5)
\]

For each word \(w\) in segment \(T_1\), we find a word in segment \(T_2\) that has the highest semantic similarity to \(w\) (\(\text{maxSim}(w, T_2)\)). Similarly, for the words in \(T_2\), we identify the corresponding words in segment \(T_1\). The similarity score of the two text segments is then calculated by combining the similarity of the words in each segment, weighted by their word specificity (i.e., IDF values).

To calculate the semantic similarity between two words \(w_1\) and \(w_2\), we use a corpus-based approach and measure the pointwise mutual information (PMI) [57, 58]:

\[
\text{PMI}(w_1, w_2) = \log_2 \frac{c(w_1 \text{ near } w_2)}{c(w_1) \cdot c(w_2)} \quad (4.6)
\]

This indicates the statistical dependency between \(w_1\) and \(w_2\), and can be used as a measure of the semantic similarity of two words. \(c(w_1 \text{ near } w_2)\) represents the number of times that word \(w_1\) appears near word \(w_2\). For this co-occurrence count, a window of length \(l\) is used, that is, we only count when \(w_1\) and \(w_2\) co-occur within this window. For a word, we define \(\text{PMI}(w, w) = 1\), therefore, \(\text{maxSim}(w, T)\) is 1 if \(w\) appears in \(T\).

The part-of-speech (POS) information of each word can also be taken into consideration when calculating the similarity of two text segments [57]. Then Equation 4.5 can be modified as:

\[
\text{maxSim}(w, T_i) = \max_{w_i \in \{T_i\}, \text{pos}(w_i) = \text{pos}(w)} \{ \text{sim}(w, w_i) \} \quad (4.7)
\]
This means when finding the \( \text{maxSim} \) between a word \( w \) and a text segment \( T_i \), we will only consider the words in \( T_i \) with the same POS as word \( w \). The reason behind this is that it is more meaningful to calculate the similarity of two words with the same POS.

Note that two different words in the two segments also contribute to the similarity score using this corpus-based approach. We call this approach corpus-based similarity following [57], even though in the cosine and centroid scores, the IDF values are also generated based on a corpus. For the MMR score, we use the corpus-based similarity for the two similarity functions \((\text{Sim}_1, \text{Sim}_2)\) in Equation 2.1, since it is more comparable than using a corpus-based similarity for \( \text{Sim}_1 \) and a cosine similarity for \( \text{Sim}_2 \).

### 4.1.3 Experimental Results and Discussion

#### Experimental Setup

The IDF values are obtained from the 69 training meetings. We split each of the 69 training meetings into multiple topics, and then use these new “documents” to calculate the IDF values. This generates more robust estimation for IDF, compared with simply using the original 69 meetings as the documents. We calculated different IDF values for the human transcripts and ASR outputs respectively, which will be used according to different transcripts. The PMI information is also generated for both the human transcripts and ASR outputs respectively.

We tagged all the meetings using the TnT POS tagger [59]. The POS model is retrained using the Penn Treebank-3 Switchboard data, which is expected to be more similar to the meeting style than domains such as Wall Street Journal.

Since the length of reference summaries varies across different annotators and meeting docu-
ments, in our experiments we generated summaries with the word compression ratio ranging from 14% to 18%.

**Experimental Results**

We evaluate the different approaches for similarity measure under the MMR framework for meeting summarization. Table 4.1 shows the summarization results (ROUGE unigram match R-1) using human transcripts for different compression ratios on the test set. The columns $Sim_1$ and $Sim_2$ are the similarity measures we used for the two similarity functions in Equation 2.1, which represent the similarity of a sentence $S_i$ to the whole document, and the similarity of the sentence $S_i$ to the currently selected summary, respectively.

Table 4.1. ROUGE-1 F-measure results (%) using different similarity approaches on test set using human transcripts.

<table>
<thead>
<tr>
<th>$Sim_1$</th>
<th>$Sim_2$</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td>cosine</td>
<td>cosine</td>
<td>65.69</td>
<td>66.23</td>
<td>66.69</td>
<td>66.70</td>
<td>66.38</td>
</tr>
<tr>
<td>centroid</td>
<td>cosine</td>
<td>68.28</td>
<td>68.48</td>
<td>69.02</td>
<td>68.99</td>
<td>68.76</td>
</tr>
<tr>
<td>corpus</td>
<td>corpus</td>
<td>68.65</td>
<td>69.16</td>
<td>68.92</td>
<td>68.54</td>
<td>68.30</td>
</tr>
<tr>
<td>corpus_pos</td>
<td>corpus_pos</td>
<td>69.66</td>
<td>69.89</td>
<td>70.08</td>
<td>69.71</td>
<td>68.99</td>
</tr>
</tbody>
</table>

Among the different similarity measures, both the centroid and the corpus-based similarity measures outperform the cosine similarity. Adding POS constraint for word similarity is also helpful, achieving the best performance among all the approaches. When POS information is considered in the corpus-based similarity measure, there is a further improvement.

Table 4.2 shows the results using ASR output on the test set. We notice that there is a performance degradation compared to using reference transcripts, but the new proposed similarity measure still outperforms the baseline. In the corpus-based method, considering POS information
does not improve the system performance, different from what have observed on the human transcript condition. This is probably because the POS tagging accuracy for the ASR transcripts is relatively low*, which impacts the word similarity in Equation 4.5.

Table 4.2. ROUGE-1 F-measure results (%) using different similarity approaches on test set using ASR output.

<table>
<thead>
<tr>
<th>Sim$_1$</th>
<th>Sim$_2$</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td>cosine</td>
<td>cosine</td>
<td>62.37</td>
<td>63.36</td>
<td>63.91</td>
<td>64.22</td>
<td>64.60</td>
</tr>
<tr>
<td>centroid</td>
<td>cosine</td>
<td>63.41</td>
<td>64.24</td>
<td>64.78</td>
<td>64.96</td>
<td>65.05</td>
</tr>
<tr>
<td>corpus</td>
<td>corpus</td>
<td>63.86</td>
<td>64.52</td>
<td>64.85</td>
<td>65.01</td>
<td>65.12</td>
</tr>
<tr>
<td>corpus_pos</td>
<td>corpus_pos</td>
<td>61.90</td>
<td>62.73</td>
<td>63.18</td>
<td>63.59</td>
<td>63.75</td>
</tr>
</tbody>
</table>

4.2 Leveraging Sentence Weights in Global Optimization Framework †

The MMR method we used in the previous section is local optimal because the decision is made based on the sentences' scores in the current iteration. In [60], the author studied modeling the multi-document summarization problem using a global inference algorithm with some definition of relevance and redundancy. The Integer Linear Programming (ILP) solver was used to efficiently search a large space of possible summaries for an optimal solution. In [27], the authors adopted this global optimization framework assuming that concepts are the minimum units for summarization, and summary sentences are selected to cover as many concepts as possible. They showed better performance than MMR. However, this method tends to select short sentences with fewer concepts in order to increase the number of concepts covered, instead of selecting sentences rich in concepts even if they overlap. According to manual examination, this seems to result in the degradation

---

*The meeting corpus is not annotated with POS information, so we cannot evaluate the POS tagging performance.
†Joint work with Benoit Favre and Dilek Hakkani-Tür
of the linguistic quality of the summary. In this section, we propose to incorporate and leverage sentence importance weights in the concept-based optimization method, and investigate different ways to use sentence weights.

### 4.2.1 Concept-based Summarization

First, we use a similar framework as in [27] to build the baseline system for this section. The optimization function is the same as Equation 2.3 and 2.4 introduced in Section 2.1.1. In our research, we use the following procedure for concept extraction, which is slightly different from the previous work in [27], where they used the rule-based algorithm for concept selection.

- Extract all content word n-grams for $n = 1, 2, 3$.
- Remove the n-grams appearing only once.
- Remove the n-grams if one of its word’s idf value is lower than a predefined threshold.
- Remove the n-grams enclosed by other higher-order n-grams, if they have the same frequency. For example, we remove “manager” if its frequency is the same as “dialogue manager”.
- Weight each n-gram $k_i$ as

$$w_i = \text{frequency}(k_i) \ast n \ast \max_j \text{idf}(\text{word}_j)$$

where $n$ is the n-gram length, and $\text{word}_j$ goes through all the words in the n-gram.

The IDF values are also calculated using the new “documents“ split according to the topic segmentation as described in Section 4.1.2. Unlike [27], we use the IDF values to remove less informative words instead of using a manually generated stopword list, and also use IDF information
to compute the final weights of the extracted concepts. Furthermore, we do not use WordNet or part-of-speech tag constraints during the extraction. Therefore, using this new algorithm, the concepts are created automatically, without requiring much human knowledge. We use this method as the baseline for our research in the section.

4.2.2 Using Sentence Importance Weight

We propose to incorporate sentence importance weights in the above summarization framework. Since the global optimization model is unsupervised, in this study we choose to use sentence weights that can also be obtained in an unsupervised fashion. We use the cosine similarity scores between each sentence and the entire document, which is also adopted as the baseline in the MMR method calculated using Equation 4.1. We investigate different ways to leverage these sentence scores in the concept-based optimization framework.

Filtering Sentences for Concept Generation

First we use sentence weights to select important sentences, and then extract concepts from the selected sentences only. The concepts are obtained in the same way as described in Section 4.2.1. The only difference is that they are generated based on this subset of sentences, instead of the entire document. Once the concepts are extracted, the optimization framework is the same as before.

Pruning Sentences from the Selection

Sentence weights can also be used to filter unlikely summary sentences and pre-select a subset of candidate sentences for summarization, rather than considering all the sentences in the doc-
ument. We use the same method to generate the summary as in Section 4.2.1, but only using preserved candidate sentences.

**Joint Optimization Using Sentence and Concept Weights**

Lastly, we extend the optimization function (Equation 2.3) to consider sentence importance weights, i.e.,

$$\text{maximize } (1 - \lambda) \sum_i w_i c_i + \lambda \sum_j u_j s_j$$  \hspace{1cm} (4.8)

where $u_j$ is the weight for sentence $j$, $\lambda$ is used to balance the weights for concepts and sentences, and all the other notations are the same as in Equation 2.3. The summary length constraint is the same as Equation 2.4. After adding the sentence weights in the optimization function, this model will select a subset of relevant sentences which can cover the important concepts, as well as the important sentences.

**4.2.3 Experimental Results**

We first use the development set to evaluate the effectiveness of our proposed approaches and the impact of various parameters in those methods, and then provide the final results on the test set.

**Baseline Results**

Several baseline results are provided in Table 4.3 using different word compression ratios for both human transcripts and ASR output on the development set. The first one (long sentence) is to construct the summary by selecting the longest sentences, which has been shown to provide...
competitive results for meeting summarization task [39]. The second one (MMR) is using cosine similarity as the similarity measure on the MMR framework. The last result (concept-based) is from the concept-based algorithm introduced in Section 4.2.1. These scores are comparable with those presented in [27]. For both human transcripts and ASR output, the longest-sentence baseline is worse than the greedy MMR approach, which, in turn, is worse than the concept-based algorithm. The performance on human transcripts is consistently better than on ASR output because of the high WER. In the following experiments, we will use the concept-based summarization results as the baseline, and a 16% word compression ratio.

Table 4.3. ROUGE-1 F-measure results (%) of three baselines on the dev set for both human transcripts (REF) and ASR output.

<table>
<thead>
<tr>
<th></th>
<th>compression</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF</td>
<td>long sentence</td>
<td>54.50</td>
<td>56.16</td>
<td>57.47</td>
<td>58.58</td>
<td>59.23</td>
</tr>
<tr>
<td></td>
<td>MMR</td>
<td>66.81</td>
<td>67.06</td>
<td>66.90</td>
<td>66.64</td>
<td>66.09</td>
</tr>
<tr>
<td></td>
<td>concept-based</td>
<td>67.20</td>
<td>67.98</td>
<td>68.30</td>
<td>67.82</td>
<td>67.51</td>
</tr>
<tr>
<td>ASR</td>
<td>long sentence</td>
<td>63.11</td>
<td>64.01</td>
<td>64.72</td>
<td>64.65</td>
<td>64.89</td>
</tr>
<tr>
<td></td>
<td>MMR</td>
<td>63.60</td>
<td>64.32</td>
<td>64.80</td>
<td>65.03</td>
<td>65.14</td>
</tr>
<tr>
<td></td>
<td>concept-based</td>
<td>63.99</td>
<td>65.04</td>
<td>65.45</td>
<td>65.44</td>
<td>65.30</td>
</tr>
</tbody>
</table>

**Filtering Sentences for Concept Generation**

In Figure 4.1, we show the results on the development set using different percentages of important sentences for concept extraction. When the percentage of the sentences is 100%, the result is the same as the baseline using all the sentences. We observe that using a subset of important sentences outperforms using all the sentences for both human transcripts and ASR output. For human transcripts, using 30% sentences yields the best ROUGE score 0.6996, while for ASR output, the best result, 0.6604, is obtained using 70% sentences.
Figure 4.1. ROUGE-1 F-measure results (%) using different percentage of important sentences during concept extraction on the dev set for both human transcripts (REF) and ASR output. The horizontal dashed lines represent the scores of the baselines using all the sentences.

**Pruning Sentences from the Selection**

This experiment evaluates the impact of using sentence weights to prune sentences and pre-select summary candidates. Figure 4.2 shows the results of preserving different percentages of candidate sentences in the concept-based optimization model. For this experiment, we use the concepts extracted from the original document. For both human transcripts and ASR output, using a subset of candidate sentences can significantly improve the performance, where the best results are obtained using 20% candidate sentences for human transcripts and 30% for ASR output. We also evaluate a length-based sentence selection and find that it is inferior to sentence score based pruning.
Figure 4.2. ROUGE-1 F-measure results (%) using pruning to preserve different percentage of candidate summary sentences on the dev set for both human transcripts (REF) and ASR output. The horizontal dashed lines represent the scores of the baselines using all the sentences.

**Joint Optimization Using Sentence and Concept Weights**

Finally we evaluate the impact of incorporating sentence scores in the global optimization framework using Equation 4.8. We use all the sentences from the documents for concept extraction and sentence selection. All sentences are weighted according to their cosine scores, and the $\lambda$ parameter is used to balance them with concept weights. Our experimental results show that sentence-level scores did not improve performance for most of the values of $\lambda$ and sometimes hurt performance. An explanation for this disappointing result is that raw sentence weights do not seem to be suitable in a global model because sentences of very different length can have similar scores. In particular, the cosine score is normalized by the total TF-IDF weight of the words of a sentence, which gives high scores to short sentences containing high-weight words. For example, if two
one-word sentences with a score of 0.9 are in the summary, they contribute 1.8 to the objective function while one two-word sentence with a better score of 1.0 only contributes 1.0 to the summary. To eliminate this problem, raw cosine scores need to be rescaled to ensure a fair comparison of sentences of different length. Therefore, in addition to using raw cosine similarity scores as the weights for sentences, we consider two variations: multiplying the cosine scores by the number of concepts and the number of words in that sentence, respectively.

Table 4.4. ROUGE-1 F-measure results (%) on the dev set for both human transcripts (REF) and ASR output.

<table>
<thead>
<tr>
<th></th>
<th>baseline</th>
<th>raw cosine</th>
<th>#concept norm</th>
<th>#words norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF</td>
<td>68.30</td>
<td>68.42</td>
<td>68.42</td>
<td>68.50</td>
</tr>
<tr>
<td>ASR</td>
<td>65.45</td>
<td>61.12</td>
<td>65.08</td>
<td>66.29</td>
</tr>
</tbody>
</table>

Table 4.4 presents results for these three methods together with the baseline scores. We can see that for human transcripts when adding cosine similarity sentence weights, the result is slightly better than the baseline. For the ASR condition, adding the cosine similarity sentence weights significantly degrades performance compared to the baseline. Reweighting the sentence scores using the number of concepts does not improve the performance; however, we observe better results by reweighting the scores based on the number of words, with more improvement on the ASR condition.

Table 4.5. ROUGE-1 F-measure results (%) of incorporating sentence importance weights on the dev set using both human transcripts (REF) and ASR output.

<table>
<thead>
<tr>
<th></th>
<th>baseline</th>
<th>sentence weights for</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>concept pruning joint</td>
<td></td>
</tr>
<tr>
<td>REF</td>
<td>68.30</td>
<td>69.96 70.17 68.50</td>
<td><strong>70.37</strong></td>
</tr>
<tr>
<td>ASR</td>
<td>65.45</td>
<td>66.04 66.51 66.29</td>
<td><strong>66.77</strong></td>
</tr>
</tbody>
</table>
Table 4.5 summarizes the results for various approaches. In addition to using each method alone, we also combine them, that is, we use sentence weights for concept extraction, and use a pre-selected set of sentences in the global optimization framework in combination with the concept scores. The best scores are obtained by combining all the proposed approaches for incorporating sentence importance weights. Among them, pruning contributes the most — using this approach alone can achieve very similar results to the best scores.

Results on Test Set

The experimental results on the test set using all the approaches proposed in this section are shown in Table 4.6. The parameter values are selected according to the performance on the dev set. The baseline results are calculated using the concept-based summarization model, obtaining comparable results to the ones presented in [27]. ROUGE-1 scores are improved using our proposed three approaches for leveraging sentence importance weights: for concept extraction, selecting the candidate summary sentences, and extending the global optimization function with reweighted sentence weights. The best results are obtained by a combination of these methods, which is consistent with our findings on the development set. The improvement is consistent for both human transcripts and ASR output. Similar patterns also hold when evaluating using ROUGE-2 (bigram) and ROUGE-SU4 (skip-bigram) scores. We also verified that the results are significantly better than the baseline according to a paired t-test ($p < 0.05$).
Table 4.6. ROUGE-1 F-measure results (%) for different word compression ratios on test set for both human transcripts (REF) and ASR output.

<table>
<thead>
<tr>
<th>compression</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>baseline</td>
<td>67.08</td>
<td>67.84</td>
<td>68.35</td>
<td>68.82</td>
<td>69.00</td>
</tr>
<tr>
<td>concept</td>
<td>68.75</td>
<td>69.80</td>
<td>70.07</td>
<td>70.24</td>
<td>69.77</td>
</tr>
<tr>
<td>pruning</td>
<td>68.85</td>
<td>69.30</td>
<td>70.10</td>
<td>70.33</td>
<td>70.43</td>
</tr>
<tr>
<td>joint</td>
<td>67.48</td>
<td>68.40</td>
<td>68.97</td>
<td>69.19</td>
<td>69.16</td>
</tr>
<tr>
<td>all</td>
<td>69.35</td>
<td>70.29</td>
<td>70.87</td>
<td>70.72</td>
<td>70.30</td>
</tr>
<tr>
<td>ASR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>baseline</td>
<td>63.30</td>
<td>64.51</td>
<td>65.31</td>
<td>65.27</td>
<td>65.84</td>
</tr>
<tr>
<td>concept</td>
<td>64.00</td>
<td>65.44</td>
<td>66.15</td>
<td>66.52</td>
<td>66.39</td>
</tr>
<tr>
<td>pruning</td>
<td>65.83</td>
<td>66.78</td>
<td>66.63</td>
<td>66.79</td>
<td>66.48</td>
</tr>
<tr>
<td>joint</td>
<td>63.82</td>
<td>64.76</td>
<td>65.80</td>
<td>66.11</td>
<td>65.77</td>
</tr>
<tr>
<td>all</td>
<td>65.87</td>
<td>66.67</td>
<td>67.07</td>
<td>67.20</td>
<td>66.91</td>
</tr>
</tbody>
</table>

4.3 Summary

In this chapter, we studied unsupervised learning approaches for extractive meeting summarization. Under the framework of MMR, we have evaluated different similarity measures. The centroid score focuses on the salient words of a text segment, ignoring words with lower TF-IDF values. The corpus-based semantic approach estimates the similarity of two segments based on their word distribution on a large corpus. Our experimental results have shown that these methods outperform the commonly used cosine similarity both on manual and ASR transcripts.

Another unsupervised approach we evaluated is a global optimization framework. Sentence level information is leveraged to improve the linguistic quality of selected summaries. First, these scores are used to filter sentences for concept extraction and concept weight computation. Second, we pre-select a subset of candidate summary sentences according to their sentence weights. Last, we extend the optimization function to a joint optimization of concept and sentence weights to cover both important concepts and sentences. Our experimental results show that these methods
can improve the system performance comparing to the concept-based optimization baseline for both human transcripts and ASR output. The best scores are achieved by combining all three approaches, which are significantly better than the baseline.
CHAPTER 5

SUPERVISED APPROACH FOR EXTRACTIVE MEETING SUMMARIZATION *

The extractive summarization task can be considered as a binary classification problem and solved using supervised learning approaches, where each training and testing instance (i.e., a sentence) is represented by a set of indicative features, and positive or negative labels are used to represent whether this sentence is a summary or not. In this chapter, we use Support Vector Machines (SVM) (the LibSVM implementation [61]) as the binary classifier because of its superior performance in many classification tasks. For each sentence in the test set, we predict its confidence score of being included into the summary. The summary is obtained by selecting the sentences with highest scores until the desired compression ratio is reached. We analyze three problems using supervised learning methods, imbalanced data problem caused by the minority of summary sentences, the human disagreement on the selection of summary sentences, and the effectiveness of different kinds of features on human transcripts and ASR output.

5.1 Features in Supervised Summarization Approaches

We extract a variety of features, including those that have been used for text and speech summarization in previous work and the new ones we propose using topic information. For now we focus on textual information and do not use acoustic or prosodic features, since the analysis and

* © 2010 Elsevier. Reprinted, with permission, from Computer Speech and Language, Improving Supervised Learning for Meeting Summarization Using Sampling and Regression, Shasha Xie and Yang Liu
methods described in this chapter are not limited to speech summarization, and can also be applied to text summarization.

5.1.1 Lexical Features†

Table 5.1 lists the lexical features. The first part includes the sentence length and the number of words in each sentence after removing the stop words. As illustrated in the example in Figure 3.1, summary sentences tend to be long, an observation similar to text summarization [62]. We also include the length information of the previous and the next sentence. Similar to [34], we use “Unigram” and “Bigram” features, which are the number of frequent words and bigrams in the sentence based on the list we automatically generated (containing words whose frequency is higher than a certain percent of the maximum frequency of all words). Finally previous work has shown that the first appearing nouns and pronouns in a sentence provide important new information [35, 63], therefore we use features to represent the number of nouns or pronouns that appear for the first time in a sentence.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Feature Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Len I, II, III</td>
<td>length of previous, current and next sentence</td>
</tr>
<tr>
<td>Num I, II, III</td>
<td># of words in previous, current, and next sentence (removing stopwords)</td>
</tr>
<tr>
<td>Unigram</td>
<td># of frequent words</td>
</tr>
<tr>
<td>Bigram</td>
<td># of frequent bigrams</td>
</tr>
<tr>
<td>Noun I, II, III</td>
<td># of first appearing nouns in previous, current and next sentence</td>
</tr>
<tr>
<td>Pronoun I, II, III</td>
<td># of first appearing pronouns in previous, current and next sentence</td>
</tr>
</tbody>
</table>

†Note that the names we use for different categories may not be perfect. But we expect that by listing all the features along with their description, it is clear about all the features used in this study.
5.1.2 Structural and Discourse Features

The structural and discourse features are described in Table 5.2. Cosine similarity between a sentence and the entire document is first included, which we also use to represent each sentence’s importance in the unsupervised framework MMR in Section 4.1.1. We derive various TF and IDF related features (e.g., max, mean, sum) for a sentence following the setup in [34, 63]. Feature “Speaker” and “Same_as_prev” are used to represent the speaker information. For each meeting, we find the most talkative speaker (who has said the most words) and speakers whose word count is more than 20% of the most talkative one. These are called main speakers. Each sentence is then labeled with whether it is said by the main speaker, and whether the speaker is the same as the previous one. To capture how term usage varies across speakers in a given meeting, we adopt the feature “SUIDF” introduced in [56]. The hypothesis for this feature is that more informative words are used with varying frequencies among different meeting participants, and less informative words are used rather consistently by different speakers.

Table 5.2. List of structural and discourse features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Feature Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosine</td>
<td>cosine similarity between the sentence and the whole document</td>
</tr>
<tr>
<td>TF I, II, III</td>
<td>Mean, Max, and Sum of TF</td>
</tr>
<tr>
<td>IDF I, II, III</td>
<td>Mean, Max, and Sum of IDF</td>
</tr>
<tr>
<td>TFIDF I, II, III</td>
<td>Mean, Max, and Sum of TF*IDF</td>
</tr>
<tr>
<td>Speaker I, II, III</td>
<td>main speaker or not for previous, current and next sentence</td>
</tr>
<tr>
<td>Same_as_prev</td>
<td>same as the previous speaker or not</td>
</tr>
<tr>
<td>SUIDF I, II, III</td>
<td>Mean, Max, and Sum of SUIDF</td>
</tr>
</tbody>
</table>
5.1.3 Topic-Related Features

Even though the meeting transcripts are not as organized as broadcast news speech (which generally consists of better story segments), they can still be divided into several parts, each with its own topic. We believe that topic segmentation contains useful information for the summarization of a meeting recording. To better capture the characteristics of different topics in a meeting, several topic related features are introduced in our study.

The topic related features we use are based on the so-called topic term frequency (TTF) and inverse topic frequency (ITF), both of which are calculated on a topic basis for each meeting transcript. The TTF is just the term frequency within a topic, and the ITF values are computed as:

\[
ITF(w_i) = \log\left(\frac{NT}{NT_i}\right)
\]

where \( NT_i \) is the number of topics containing word \( w_i \) within a meeting, and \( NT \) is the total number of topic segments in this meeting. Note that ITF values are estimated for each meeting, whereas the IDF values in the structural and discourse feature set (Table 5.2) are calculated based on the entire corpus. Our hypothesis is that this meeting specific ITF might be more indicative of a specific topic in this meeting. Table 5.3 shows the topic related features we used.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Feature Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( ITF I, II, III )</td>
<td>Mean, Max, and Sum of ITF</td>
</tr>
<tr>
<td>( TTFITF I, II, III )</td>
<td>Mean, Max, and Sum of TTF*ITF</td>
</tr>
</tbody>
</table>
5.2 Improving Supervised Summarization Approaches

5.2.1 Issues in Supervised Learning for Meeting Summarization

This chapter aims to investigate the following three issues in supervised learning for meeting summarization. First, since summary sentences are a small percent of the original documents, there is an imbalanced data problem. In the ICSI meeting corpus, the average percent of the positive samples is 6.62%. When learning from such imbalanced data sets, the machine learning models tend to produce high predictive accuracy over the majority class, but poor predictive accuracy over the minority class [64]. Different methods have been proposed in the machine learning community for this problem, such as up-sampling and down-sampling, both aiming to make the data more balanced for classifier training. However, this problem has never been studied for the meeting summarization task. We propose different approaches to deal with the imbalanced data problem by utilizing the original annotated data.

Second, we notice that human annotators often do not agree with others in the selection of summary sentences [54]. In the training set we use, there is only one human annotation available. Because of the large variation in human annotation, a non-summary sentence may be similar to an annotated summary sentence, and other annotators may select this sentence in the summary if multiple annotations were available. We believe these negative samples are noisy and may affect the classifiers to effectively learn to distinguish the two classes. In Table 5.4, we show two similar sentences in one meeting transcript from the ICSI meeting corpus, where the first sentence was selected by the human annotator to be in the summary. When using binary classification for this task, this kind of labeled data is likely to introduce noise and may be misleading for the classifier.
Table 5.4. Example of two similar sentences with different labels.

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>i think for word-level this would be ok.</td>
<td>+1</td>
</tr>
<tr>
<td>for word-level it’s alright.</td>
<td>-1</td>
</tr>
</tbody>
</table>

This problem can be solved by different approaches. The first one is in line with the sampling methods that are motivated to address the imbalanced data problem as mentioned earlier. We reconstruct the training samples to reduce the effect of these confusing instances, either by changing their labels from negative to positive, or removing them from the training set. This sampling method increases the positive to negative ratio of the training set. Changing the labels of instances from negative to positive is an idea similar to up-sampling that increases the number of the positive instances and reduces the number of negative ones; removing the misleading instances from the negative class is a down-sampling method. The difference between our proposed approaches and the traditional sampling methods is that we will focus on the confusable negative examples and thus we expect this will at the same time address the human annotation disagreement problem.

The second approach we suggest is to reframe the summarization task using a regression model instead of binary classification, where we assign each non-summary training sentence a numerical weight according to its similarity to the labeled summary sentences. These weights can provide more elaborate information for the learning models.

Third, we evaluate the effectiveness of the features introduced in Section 5.1. Some prior work has evaluated the feature usage for this task (e.g., [32]). Here we use forward feature selection (FFS) to select a subset of important features, and compare their difference on human transcripts and ASR output.
5.2.2 Addressing the Imbalanced Data Problem

The summary sentences are much fewer than the non-summary sentences for a meeting transcript, thus there is an imbalanced data problem for the summarization task. The classification performance often degrades when faced with the imbalanced class distributions [65]. Most of the classification algorithms are developed to maximize the classification accuracy; however, when the class distribution is imbalanced, the classifier can still achieve a high accuracy even though it fails to detect or classify the minority class (which is often the more important class for most tasks). A common practice for dealing with imbalanced data sets is to rebalance them artificially using "up-sampling" (e.g., replicating instances from the minority class) and "down-sampling" (selecting some samples from the majority class). In addition to modifying the data distribution, it is also possible to modify the classifier [66]. In [67], Liu et al. investigated the use of different sampling approaches for the task of sentence boundary detection in speech. For the task of extractive summarization, we propose to deal with this problem by reselecting the training instances and increasing the positive to negative ratio. We evaluated three approaches: up-sampling, down-sampling and re-sampling.

Up-Sampling

The goal of up-sampling is to increase the number of positive samples. A simple way for this is to replicate the positive samples in the training set. In our approach, we achieve up-sampling by selecting the negative samples that are most similar to the positive ones and changing their labels from negative to positive.

In order to find out the confusable samples, we will assign a weight for each non-summary sen-
tence, which measures its similarity to the reference summary sentences. We then select the sentences with high weights and change their labels to positive. This reduces the number of negative samples, increases the positive samples, and thus increases positive to negative ratio. Varying the number of the selected sentences will result in different positive to negative ratios. In Figure 5.1, we illustrate how up-sampling changes the labels of the instances and the decision boundaries. Figure 5.1(a) shows the original distribution and the hyperplane in SVM for separating the positive and negative instances. After up-sampling, the negative samples that are similar to the positive samples (likely to be close to the decision boundary or in the decision region for the positive class) are changed to positive ones. The hyperplane is moved accordingly, as shown in Figure 5.1(b). We will evaluate the impact of the number of the negative samples that are selected for a label change. If few samples are selected, the effect of the noisy instances may still be there; if too many instances are selected, additional noise may be introduced because some unimportant sentences are marked as summary sentences.

![Figure 5.1](image.png)

(a) Original distribution  
(b) After up-sampling

Figure 5.1. Illustration of up-sampling for binary classification.

In this up-sampling method, for each non-summary sentence in a training transcript, we want
to assign a weight to indicate its similarity to the reference summary of the transcript. We examine different similarity measures: cosine similarity (as shown in Eq 1) and ROUGE score (ROUGE-1 F-measure). For each of the two measures, we use the score of the sentence to the entire reference summary, as well as the maximum and mean value of the similarity scores with individual summary sentences. The two methods, cosine similarity and ROUGE are similar in the sense that they both measure the word match (counting the matched words in a test sentence and reference summary sentences), but they use different measurement: cosine score normalizes the matches (dot product) by the product of the length of the two vectors; ROUGE score is the harmonic mean of the precision and recall (number of matched words normalized by the length of each individual vector). In addition, for the cosine similarity measure, TFIDF values are used as the term weights, whereas IDF information is not used in ROUGE. Since ROUGE is the final evaluation metric for summarization, we expect that it might be a better similarity measure for sampling. In total, we have 6 different weighting scores, as listed in Table 5.5. Their impact will be compared in Section 5.3.2.

Table 5.5. Weighting measures used in sampling for non-summary sentences.

<table>
<thead>
<tr>
<th>Weighting Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosine</td>
<td>ALL cosine similarity to the entire reference summary</td>
</tr>
<tr>
<td></td>
<td>MAX, MEAN max or mean of cosine similarity to each reference summary sentence</td>
</tr>
<tr>
<td>ROUGE</td>
<td>ALL ROUGE score to the entire reference summary</td>
</tr>
<tr>
<td></td>
<td>MAX, MEAN max or mean of ROUGE scores to each reference summary sentence</td>
</tr>
</tbody>
</table>
Down-Sampling

The goal of down-sampling is to decrease the number of negative samples for an increased positive to negative ratio. Instead of using random down-sampling (selecting negative samples randomly), we want to choose the negative samples that are most likely to be confusable with positive examples and remove those from training, such that the two classes are more separable. We use the same weighting measures for the negative instances as in up-sampling and select the ones with high weights for removal according to different sampling ratio. Figure 5.2 illustrates the distribution difference after down-sampling. Because some of the negative instances close to the decision boundary are removed, the new hyperplane now can better classify the remaining instances in the data. For example, the positive instance closest to the boundary in Figure 5.2(b) was labeled as negative by the original classifier, but is now correctly classified using the new model.

Note that the down-sampling and up-sampling methods above also account for the human annotation disagreement, for example, a negative instance will be relabeled as a positive one if it is similar enough to the positive samples. This is likely to be the case if multiple human annotations were available.

Re-Sampling

The re-sampling method we propose can be considered as a two-pass processing as shown in Figure 5.3. A weighting method is applied to select a subset of training data, and only these selected samples are used for training the classifier. For testing, we first select some samples as the candidate summary sentences, and then supervised learning method is only applied on these
selected subset using the classifier trained before. The final summary is generated according to the confidence scores for each candidate sentence.

In this method, a salience score is needed for each sentence, in both positive and negative classes, and for both training and testing. Because this score is also applied for testing, we cannot use the weighting measures shown in Table 5.5, which need the reference summary for the similarity computation. Therefore we propose two new methods for computing the sentence salience score: one is based on the sum of the TFIDF values of the words in the sentence; the other is the cosine similarity of the sentence and the entire document. The main difference between these two methods is that cosine similarity is normalized by the sentence length. These two methods can be computed without any information of the labeled summary for a given training or testing transcript. In general, both of these two weighting methods give higher scores to summary sentences than non-summary ones. In fact, these two methods are often used in unsupervised extractive summarization to select summary sentences. Using these salience scores, we select a subset of the sentences with higher weights as the instances for training, or the candidates for testing.
Figure 5.3. Flowchart for re-sampling method.

This re-sampling technique helps address the imbalanced data problem in training. It is supposed to preserve most of the positive instances and remove negative instances, thus increasing the positive to negative ratio. In order to verify that after removing the sentences with lower weights, the remaining samples still include most of the positive samples, and the training data is more balanced, we calculate the average coverage of the original positive sentences and the percentage of positive instances after re-sampling using different sampling rates for the training data. Results are demonstrated in Figure 5.4. We can see that the top 50% sentences can preserve 94.3% of the positive sentences when using TFIDF scores as the selection criteria. The positive percentage after re-sampling is much higher than that in the original data (6.62%). Figure 5.4 also shows that TFIDF scores outperform cosine similarity on the coverage of positive samples or the percentage of positive sentences. In terms of the negative samples removed from the training data, this re-sampling method is another down-sampling approach. Unlike the down-sampling we propose
in previous section, this re-sampling removes instances that are further away from the decision boundary (as they have low similarity scores to the entire document).

During testing, this re-sampling approach only keeps those sentences with high weights and reduces the number of candidates samples. Since most of the sentences ignored are non-summary ones, this does not have a negative impact, but rather allowing the model to focus on the more likely candidate sentences.

![Figure 5.4](image)

Figure 5.4. Coverage of the original positive samples (left Y-axis) and the percentage of positive samples in the selected data (right Y-axis) using TFIDF and cosine scores as selection criteria for different re-sampling rates.

### 5.2.3 Using Regression Model for Summarization

Another problem using statistical learning for meeting summarization is that it may not be optimal to treat the summarization task as a binary classification problem — two similar sentences
may be annotated with two different labels because of the preference by different annotators or the need to avoid redundancy for summary selection or other reasons, as shown in the example in Table 5.4. With such kind of confusion it may be hard for the model to learn to separate summary and non-summary sentences. Therefore, instead of using binary labels, we hypothesize that the summarization task can be modeled as a regression problem, where the labels are numerical numbers representing the importance of the sentences. We expect that the fine-grained weights can provide more information of each sentence’s significance and help train a more discriminative model.

The idea of assigning salience weight for the training instances is similar to the weighting methods we used for up-sampling and down-sampling. We keep the label 1 for the summary sentences, and compute target labels for those non-summary sentences using the same 6 weighting measures listed in Table 5.5. Table 5.6 shows the numerical labels using \textit{cosine_max} for the example sentences used in Table 5.4. The non-summary sentence has a negative label originally, but now is assigned a high target label because of its similarity to the reference summary sentence. For comparison, we also include the labels for this same example using up-sampling (changed to positive) and down-sampling (this instance is removed).

Table 5.6. Example of new labels using regression, up-sampling, and down-sampling for a non-summary sentence (second row) that is similar to a summary sentence (first row).

<table>
<thead>
<tr>
<th>Sentence</th>
<th>original</th>
<th>regression</th>
<th>up-sampling</th>
<th>down-sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>i think for word-level this would be ok.</td>
<td>1</td>
<td>1.0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>for word-level it’s alright.</td>
<td>-1</td>
<td>0.968</td>
<td>1</td>
<td>removed</td>
</tr>
</tbody>
</table>

Once the target labels are assigned to all the training instances, we will use a regression model
to learn the underlying function to estimate the target labels. Regression analysis is a statistical tool for investigating relationships between variables. A simple regression model is linear regression that makes the prediction of one variable based on the knowledge of another one when there is a statistically significant correlation between the two variables [68]. Another regression model is logistic regression, which is used for predicting the probability of occurrence of an event by fitting data to a logistic curve [69]. This model can be used as a classifier too, where during training, the instances have two classes, and during testing, the model provides a posterior probability of the membership of the test instance. However, our approach here is to use actual target label for each training instances rather than the original binary labels.

The Support Vector Regression is the regression model we use in this work [70]. Similar to SVM for binary classification, the goal of SVR is to find a function $f(x)$ that has at most $\epsilon$ deviation from the actual targets $y_i$ for all the training instances $x_i$, and at the same time is as flat as possible. Our preliminary experiments showed that SVR outperformed the logistic regression or neural network. During testing, a regression score is predicted for each testing sample, then we use the same method as in the classification approach to select the sentences based on their confidence scores.

### 5.2.4 Forward Feature Selection

No matter which model we use for the summarization task, whether it is binary classification or regression, the training and testing samples are represented by the feature set described in Section 5.1. In our study, we use forward feature selection (FFS) [71] to analyze the importance of different features. The algorithm of FFS is shown below in Algorithm 1. It is a greedy algorithm. In every
iteration step, the best feature is selected when its addition to the current feature subset leads to the greatest performance improvement.

**Algorithm 1** Algorithm for Forward Feature Selection

Let $P = \emptyset$ be the current set of selected features

Let $Q$ be the full set of features

while size of $P$ is smaller than a given constant do

(a) for each $v \in Q$

Set $P' \leftarrow P \cup \{v\}$

Train the model with $P'$ and evaluate on the dev set

(b) Set $P \leftarrow P \cup \{v^*\}$

where $v^*$ is the the best feature obtained in step (a)

(c) Set $Q \leftarrow Q \setminus \{v^*\}$

(d) Record performance using current $P$

end while

5.3 Experimental Results and Discussion

5.3.1 Baseline Results

We provide two baseline results for comparison. The first one generates a summary by selecting the longest sentences until reaching the specified length. The second one is the supervised approach that selects the sentences with high confidence scores predicted by the SVM model using all the features we described in Section 5.1. We provide results for a few different compression ratios varying from 13% to 18%.

<table>
<thead>
<tr>
<th>compression ratio</th>
<th>13%</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long sent selection</td>
<td>52.38</td>
<td>54.50</td>
<td>56.16</td>
<td>57.47</td>
<td>58.58</td>
<td>59.23</td>
</tr>
<tr>
<td>All features</td>
<td>67.25</td>
<td>67.80</td>
<td>67.76</td>
<td>67.56</td>
<td>67.22</td>
<td>66.86</td>
</tr>
</tbody>
</table>
Table 5.8. ROUGE-1 F-measure results (%) of the baselines using ASR outputs.

<table>
<thead>
<tr>
<th>compression ratio</th>
<th>13%</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long sent selection</td>
<td>62.13</td>
<td>63.11</td>
<td>64.01</td>
<td>64.72</td>
<td>64.65</td>
<td>64.89</td>
</tr>
<tr>
<td>All features</td>
<td>62.05</td>
<td>63.35</td>
<td>64.24</td>
<td>64.64</td>
<td>64.77</td>
<td>64.38</td>
</tr>
</tbody>
</table>

Table 5.7 and Table 5.8 show the ROUGE-1 F-scores for the two baseline systems on the dev set using the human transcripts and ASR output, respectively. Using sentence length to select summary yields worse performance than using the classification approach on the human transcript condition; however, the two systems achieve similar results on the ASR output. This finding is consistent with the results reported in [39]. Comparing the performance on the human transcripts and ASR output using classification approach, we see that the results are consistently better for human transcripts on different compression ratios, which is expected. We can also compare the scores with the ones in Table 4.3 obtained using the unsupervised approaches, MMR and the concept-based framework, introduced in Section 4.2.3. This supervised baseline outperforms the MMR method, and achieved comparable results as the concept-based one when using human transcripts. Overall, the baseline results are very competitive with the previous work [13, 34, 39]. For the following experiments, we use the results achieved by using the SVM classifier with all the features as the baseline, and evaluate the performance improvement using our proposed approaches.

5.3.2 Results Using Sampling Methods

Since our focus here is on the approaches to deal with the imbalanced data problem, we fix the word compression ratio and evaluate the effect of different sampling rates and weighting methods on summarization performance using the development set. We use 14% and 17% word compres-
sion ratio for the human transcript and ASR outputs respectively. These compression ratios are chosen based on the baseline results above.

**Experimental Results Using Up-Sampling**

Up-sampling is achieved by changing the labels of some negative instances to positive. These negative instances are selected based on their similarity to the positive samples. In Section 5.2.2, we described 6 weighting methods to assign scores to the negative instances for up-sampling, 3 based on cosine similarity, and the other 3 using ROUGE scores. Figure 5.5 shows the ROUGE-1 F-scores using the 6 weighting methods and different up-sampling rates on the human transcript for the development set. The X-axis, the sampling rate, is the rate of the current positive samples to the original positive instances. When it is 1, none of the negative instances is changed to positive and there are no newly added positive samples, that is, the results are the same as the baseline system. We can see that different similarity measures and up-sampling ratios have great influence on the system performance. Of all the weighting methods, the best results are obtained using ROUGE\_mean and increasing the positive samples to 1.5 times of the original number. Compared to the baseline result, up-sampling yields an improvement from 67.80% to 69.24%. Using the similarity measure ROUGE\_mean, the performance improves when we change more negative instances to positive until the sampling ratio reaches 1.5. After that, increasing the up-sampling ratio leads to performance drop. But for other similarity measures, the trend is not clear — the results are more random and there is more fluctuation.

The results for the ASR condition are shown in Figure 5.6. The same setting, ROUGE\_mean and 1.5 up-sampling rate, also outperforms the baseline result (65.66% vs. 64.77%). However,
Figure 5.5. ROUGE-1 F-measure results (%) using up-sampling on human transcripts. using the weighting measure Cosine\_mean and a sampling rate of 4, achieves slightly better result, 65.97%. But the pattern using different sampling rates for the Cosine\_mean similarity measure is not as clear as for ROUGE\_mean. Comparing ASR and human transcripts, it seems that a higher up-sampling rate is often preferred for ASR output and there is more fluctuation in the human transcript condition when varying the sampling rates.

Our proposed up-sampling approach is different from commonly used up-sampling method that replicates the samples in the minority class, therefore next we evaluate the performance of up-sampling by varying the number of times that we replicate the summary sentences. The ROUGE-1 F-measure results from these experiments are shown in Table 5.9 using the human transcripts and ASR output respectively. When the up-sampling rate is 1, it is the same as the baseline setup. For human transcripts, replicating the positive class degraded performance; in contrast, our proposed up-sampling method can yield performance gain. When using ASR output, for some upsampling
rates, there is an improvement. The best result is similar to that in our proposed up-sampling method. However, there can also be significant performance drop for some up-sampling rates. Compared to our approach above, the performance variance when changing the up-sampling rates seems to be greater when up-sampling is achieved by replicating minority samples. Overall, for all the up-sampling approaches, there is not a consistent correlation between the system performance and how balanced the resulting data set is after up-sampling.

Table 5.9. ROUGE-1 F-measure results (%) of up-sampling by replicating the positive samples on both human transcripts and ASR outputs.

<table>
<thead>
<tr>
<th>Up-sampling rate</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>67.80</td>
<td>63.73</td>
<td>57.39</td>
<td>55.18</td>
<td>57.89</td>
<td>58.36</td>
</tr>
<tr>
<td>ASR</td>
<td>64.77</td>
<td>59.94</td>
<td>66.18</td>
<td>65.57</td>
<td>65.49</td>
<td>66.04</td>
</tr>
</tbody>
</table>
Experimental Results Using Down-Sampling

The same weighting methods are used for down-sampling, with the goal of removing negative instances with high similarity scores to the summary sentences. Figure 5.7 shows the down-sampling results on the development set using human transcripts. The X-axis, the down-sampling rate, represents the percentage of the removed negative instances. When it is 0, it means that no negative instances are removed, which is the baseline setup. We see from the results that the best performance is obtained using \textit{ROUGE\_mean} as the weighting method with a down-sampling rate of 5\%. The performance is improved from 67.80\% to 70.28\%. The experimental results on the ASR outputs are shown in Figure 5.8. Using the same weighting method and down-sampling rate, we obtain the best score for ASR outputs, 66.42\% compared to the baseline score of 64.77\%. For this similarity measure \textit{ROUGE\_mean}, we observe that the best result is achieved with a sampling rate of 5\%, then the performance starts degrading when removing more instances from the data set. This observation is consistent for the human transcripts and ASR output.

In up-sampling and down-sampling, for most of the experiments, \textit{ROUGE\_mean} outperforms the other weighting measures, except for up-sampling on ASR output, where \textit{Cosine\_mean} is slightly better. This is consistent with our expectation that ROUGE is the final evaluation metric, and is expected to be a better weighting method for capturing the sentence similarity. The best scores are obtained by the mean value of the weighting methods, which suggests that the selected sentences should be the most similar ones to the entire summary, not to a specific sentence (as is done using the max value). In addition, the fact that the average of the cosine similarity or ROUGE scores yields better performance indicates that it is better to give equal weight to different summary sentences than more weight to longer sentences.
Figure 5.7. ROUGE-1 F-measure results (%) using down-sampling on human transcripts.

Figure 5.8. ROUGE-1 F-measure results (%) using down-sampling on ASR outputs.
For a comparison with the down-sampling approach above, we also evaluate a commonly used down-sampling method, e.g., randomly removing the negative samples from the data set. Because the removed samples are randomly selected, we performed three random sampling and obtained the average results of the three runs. Table 5.10 shows the ROUGE-1 results for human transcripts and ASR output respectively when varying the down-sampling rates. Using the human transcripts, there is only marginal change of the results for different sampling rates. The results on ASR output fluctuate for different sampling rates. There is performance gain for a few different setups. Our proposed approach outperforms this commonly used down-sampling method for both human transcripts and ASR condition. In addition, our approach has similar patterns on the human transcripts and ASR output, whereas this down-sampling by random selection has a different trend for these two conditions.

Table 5.10. ROUGE-1 F-measure Results (%) of down-sampling by randomly removing negative samples on both human transcripts and ASR output.

<table>
<thead>
<tr>
<th>Down-sampling rate (%)</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>68.53</td>
<td>67.86</td>
<td>67.86</td>
<td>68.34</td>
<td>68.33</td>
<td>68.00</td>
</tr>
<tr>
<td>ASR</td>
<td>64.74</td>
<td>65.30</td>
<td>64.31</td>
<td>65.35</td>
<td>66.10</td>
<td>64.52</td>
</tr>
</tbody>
</table>

**Experimental Results Using Re-Sampling**

For re-sampling, we use different selection criteria to retain some training and test instances. We compare two measurements: TFIDF and cosine similarity, as introduced in Section 5.2.2. Figure 5.9 shows the re-sampling results on human transcripts and ASR outputs for the development set. The X-axis, re-sampling rate, is the percentage of the instances preserved. When it is 100%, we do not delete any instances, so the results are the same as the baseline. For human transcripts,
we notice that when using cosine similarity scores to keep the top 20% instances, we obtain the best score of 70.41%. The sampling rate is relatively low (20%), which results in an average positive coverage of 39.3% on the training set. This implies that the positive coverage is not the only criteria to evaluate a weighting method for re-sampling. The system performance is also dependent on which sentences are actually preserved by this weighting method, for both positive and negative classes. From the results on ASR outputs, we observe very different patterns. The best score on ASR outputs is obtained using TFIDF as the selection metric, and keeping the top 35% samples. This yields a performance improvement from 64.77% to 66.27%.

Figure 5.9. ROUGE-1 F-measure results (%) using the re-sampling method.

5.3.3 Regression Results

To use the regression model, we need continuous labels for the training instances. These target labels are generated using the same weighting methods as used for up-sampling and down-sampling. The regression results on human transcripts and ASR outputs are shown in Figure 5.10 for different word compression ratios. The baseline result is from the binary classification sys-
tem using all the features. Here we only present results for the best weighting measure using cosine similarity and ROUGE. Using human transcripts, we notice that the results of cosine-based weighting methods are worse than the baseline system, however, using ROUGE-based weighting, we obtain better performance, for example, 69.23% for the word compression ratio of 14%. The patterns for ASR outputs are different from the human transcripts. Cosine-based weighting is superior to ROUGE-based measures, and both achieve better performance than the baseline.

![Graphs showing ROUGE-1 F-measure results (%) using the regression model.](image)

(a) on human transcripts  
(b) on ASR outputs

Figure 5.10. ROUGE-1 F-measure results (%) using the regression model.

Then we evaluate the feasibility of combining regression and sampling approaches. For each sampled instance, we change its binary label to a continuous number in order to use a regression model for training and testing. Since the up-sampling approach selects a negative sentence and changes its label to positive, when assigning continuous weights for each instance, there is no difference with just using the regression setup itself. However, for down-sampling and re-sampling methods, the results are different because some of the instances are removed from the corpus. Therefore we only combine regression with down-sampling and re-sampling. Table 5.11 shows the combination results, along with using each method alone. We observe that combining these
two methods can not yield further improvement. This is probably because the sampling methods are proposed mainly to deal with the imbalanced data problem, which does not exist in regression model. The labels of the instances in regression are numerical numbers, and the summarization task is solved using a regression model instead of a binary classification model.

Table 5.1. ROUGE-1 F-measure results (%) of combining sampling and regression. ‘Ds’ is down-sampling; ‘Rs’ is re-sampling; ‘Rg’ is regression.

<table>
<thead>
<tr>
<th></th>
<th>baseline</th>
<th>Ds</th>
<th>Rs</th>
<th>Rg</th>
<th>Rg&amp;Ds</th>
<th>Rg&amp;Rs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Human</strong></td>
<td>67.80</td>
<td>70.28</td>
<td>70.41</td>
<td>69.23</td>
<td>68.99</td>
<td>69.04</td>
</tr>
<tr>
<td><strong>ASR</strong></td>
<td>64.77</td>
<td>66.42</td>
<td>66.27</td>
<td>66.59</td>
<td>66.56</td>
<td>66.56</td>
</tr>
</tbody>
</table>

### 5.3.4 Feature Selection Results

All the instances in previous experiments are represented by the features described in Section 5.1. In order to analyze the effectiveness of different feature subsets, we use the forward feature selection (FFS) algorithm in Section 5.2.4 to rank each feature.

**Features Selected Using FFS**

We run forward feature selection on the development set. The original data set (i.e., without any sampling or regression) is used first to select features. Instead of using classification accuracy or error rate, we use the ROUGE score (ROUGE-1 F-measure) as the selection criterion, since that is our ultimate performance measure. We train an SVM model, and predict each sentence’s confidence score of being in the summary on the development set. The summary is extracted by selecting the sentences with the highest probabilities, with a word compression ratio of 14% for human transcripts, and 17% for ASR output.
The top features selected incrementally by FFS for human transcripts are listed in Table 5.12, along with the ROUGE-1 F-measure score after adding each feature iteratively. The score when using all the features is also shown in the table for a comparison. The results show that a subset of the features can outperform using all the features, therefore feature selection is important for this task. The sentence length is selected as the top feature, which also validates our choice of the baseline system by selecting long sentences. The frequent unigram, bigram and the first appearing noun and pronoun are also included in the top features, which shows that the lexical cues are very predictive in the domain of meeting summarization. Different TF-IDF related features, such as $\text{IDF I and II, TF III, TFIDF II and III}$, prove to be very helpful for summarization, which is consistent with the previous research [34, 56, 63]. In addition, feature $\text{SUIDF III}$, representing the term usage in different speakers suggested in [56], also proves to be useful for this task.

Table 5.12. ROUGE-1 F-measure results (%) after Forward Feature Selection for human transcripts.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Feature Description</th>
<th>ROUGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram</td>
<td>most frequent bigram</td>
<td>68.92</td>
</tr>
<tr>
<td>+ IDF I</td>
<td>Mean of IDF</td>
<td>70.12</td>
</tr>
<tr>
<td>+ TFIDF II</td>
<td>Max of TFIDF</td>
<td>70.67</td>
</tr>
<tr>
<td>+ Noun II</td>
<td># of first appearing noun in current sentence</td>
<td>70.18</td>
</tr>
<tr>
<td>+ Len I</td>
<td>previous sentence length</td>
<td>69.91</td>
</tr>
<tr>
<td>+ Unigram</td>
<td>most frequent words</td>
<td>69.85</td>
</tr>
<tr>
<td>+ IDF II</td>
<td>Max of IDF</td>
<td>70.49</td>
</tr>
<tr>
<td>+ Len II</td>
<td>current sentence length</td>
<td>71.09</td>
</tr>
<tr>
<td>+ Noun I</td>
<td># of first appearing noun in previous sentence</td>
<td>71.35</td>
</tr>
<tr>
<td>+ TF III</td>
<td>Sum of TF</td>
<td>71.45</td>
</tr>
<tr>
<td>+ SUIDF III</td>
<td>Sum of SUIDF</td>
<td>71.09</td>
</tr>
<tr>
<td>+ Noun III</td>
<td># of first appearing noun in next sentence</td>
<td>70.85</td>
</tr>
<tr>
<td>+ Cosine</td>
<td>cosine similarity score</td>
<td>70.85</td>
</tr>
<tr>
<td>+ TFIDF III</td>
<td>Sum of TFIDF</td>
<td>71.18</td>
</tr>
<tr>
<td>+ Pronoun II</td>
<td># of first appearing pronoun in current sentence</td>
<td>71.55</td>
</tr>
<tr>
<td>All features</td>
<td></td>
<td>67.80</td>
</tr>
</tbody>
</table>
The features selected for ASR output are shown in Table 5.13. Fewer features are selected comparing with the features selected for human transcripts. The structural feature (TFIDF) is very important for ASR condition, similar to human transcripts. However, speaker information contributes more on the ASR condition, where two speaker related features are selected: Speaker I and III. The topic related feature, ITF II, that we have introduced, is also selected. Surprisingly, the sentence length feature is not selected, even though we have seen from Table 5.8 that selecting summary sentences simply based on length achieves very good performance. This is possibly because the feature selection process is a greedy algorithm.

The feature selection results on both human transcripts and ASR output show that using one single feature can yield better performance than using all the features. In addition, we performed FFS using the regression model as the underlying classifier instead of the binary classifier, however, it did not yield any gain compared to using all the features.

Table 5.13. ROUGE-1 F-measure results (%) after Forward Feature Selection for ASR outputs.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Feature Description</th>
<th>ROUGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFIDF III</td>
<td>Sum of TFIDF</td>
<td>65.82</td>
</tr>
<tr>
<td>+ ITF II</td>
<td>Max of ITF</td>
<td>66.29</td>
</tr>
<tr>
<td>+ Speaker III</td>
<td>next sentence said by main speaker or not</td>
<td>67.20</td>
</tr>
<tr>
<td>+ Speaker I</td>
<td>previous sentence said by main speaker or not</td>
<td>67.28</td>
</tr>
<tr>
<td>All features</td>
<td></td>
<td>64.77</td>
</tr>
</tbody>
</table>

Using Selected Feature with Sampling and Regression

Our next work is to answer whether feature selection is complementary to the sampling and regression approaches. We use the selected features together with the best setup from sampling (e.g., the weighting measure and the sampling rate) and regression (e.g., the weighting measure)
based on the experimental results using all the features in Section 5.3.2. Table 5.14 and 5.15 show the ROUGE-1 results for the human transcripts and ASR respectively. The baseline is the results obtained using all the features. The second row shows the results using the top features selected by FFS. For both human transcripts and ASR output, using the selected subset of features outperforms the baseline for different compression ratios. However, when representing the training and testing instances using these selected features for sampling and regression, there is no further improvement, even though several results are better than the baseline. Overall, the sampling results are better than regression, which is probably because the feature selection is conducted under a binary classification assumption. These results demonstrate that these different methods are not complementary. Feature selection is closely related with the task definition, the objective function, and the base classifier. We need to pay attention if we want to use the selected features for further applications.

Table 5.14. ROUGE-1 F-measure results (%) using selected features and sampling and regression on human transcripts.

<table>
<thead>
<tr>
<th>compression ratio</th>
<th>13%</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline (all features)</td>
<td>67.25</td>
<td>67.80</td>
<td>67.76</td>
<td>67.56</td>
<td>67.22</td>
<td>66.86</td>
</tr>
<tr>
<td>selected features</td>
<td>71.53</td>
<td>71.55</td>
<td>71.15</td>
<td>70.48</td>
<td>69.58</td>
<td>68.51</td>
</tr>
<tr>
<td>selected features</td>
<td>67.37</td>
<td>67.89</td>
<td>67.88</td>
<td>67.38</td>
<td>66.88</td>
<td>66.40</td>
</tr>
<tr>
<td>Up-sampling</td>
<td>70.27</td>
<td>70.55</td>
<td>70.60</td>
<td>70.03</td>
<td>69.37</td>
<td>68.41</td>
</tr>
<tr>
<td>Down-sampling</td>
<td>69.86</td>
<td>70.47</td>
<td>70.23</td>
<td>69.93</td>
<td>69.33</td>
<td>68.68</td>
</tr>
<tr>
<td>Re-sampling</td>
<td>67.07</td>
<td>67.24</td>
<td>66.87</td>
<td>66.35</td>
<td>65.91</td>
<td>65.46</td>
</tr>
<tr>
<td>Regression</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.3.5 Experimental Results on Test Set

In this section, we show the experimental results on the test set for all the proposed methods we introduce in this chapter. All of the parameters and setups are chosen based on the results on
Table 5.15. ROUGE-1 F-measure results (%) using selected features and sampling and regression on ASR output.

<table>
<thead>
<tr>
<th>compression ratio</th>
<th>13%</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline (all features)</td>
<td>62.05</td>
<td>63.35</td>
<td>64.24</td>
<td>64.64</td>
<td>64.77</td>
<td>64.38</td>
</tr>
<tr>
<td>selected features</td>
<td>64.99</td>
<td>65.90</td>
<td>66.69</td>
<td>67.36</td>
<td>67.28</td>
<td>67.03</td>
</tr>
<tr>
<td>selected features</td>
<td>Up-sampling</td>
<td>65.86</td>
<td>66.28</td>
<td>66.78</td>
<td>66.71</td>
<td>66.67</td>
</tr>
<tr>
<td>Down-sampling</td>
<td>65.36</td>
<td>66.20</td>
<td>66.29</td>
<td>66.54</td>
<td>66.56</td>
<td>66.00</td>
</tr>
<tr>
<td>Re-sampling</td>
<td>58.59</td>
<td>59.87</td>
<td>60.95</td>
<td>61.85</td>
<td>62.29</td>
<td>62.73</td>
</tr>
<tr>
<td>Regression</td>
<td>65.57</td>
<td>66.27</td>
<td>66.69</td>
<td>66.39</td>
<td>66.15</td>
<td>65.92</td>
</tr>
</tbody>
</table>

The development set. Table 5.16 includes the results for human transcripts. The different sampling methods and regression model improve the performance, with the best scores from down-sampling. The results on ASR outputs are shown in Table 5.17. We notice performance gain from all of the proposed methods. Unlike on the human transcripts, up-sampling and regression yield the best performance among all the approaches for the ASR condition.

We also run 10-fold cross validation on the training set. The results are consistent with those on the test set, where down-sampling achieves the best performance on human transcripts, and up-sampling and regression outperform other approaches on ASR output. The improvement from these methods compared to the baseline results is statistically significant using a paired t-test ($p < 0.05$).

Table 5.16. ROUGE-1 F-measure results (%) on test set for human transcripts.

<table>
<thead>
<tr>
<th>compression ratio</th>
<th>13%</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline (all features)</td>
<td>68.29</td>
<td>69.15</td>
<td>69.71</td>
<td>69.83</td>
<td>69.91</td>
<td>69.78</td>
</tr>
<tr>
<td>all features</td>
<td>Up-sampling</td>
<td>69.82</td>
<td>70.31</td>
<td>70.39</td>
<td>70.29</td>
<td>69.99</td>
</tr>
<tr>
<td>Down-sampling</td>
<td>70.13</td>
<td>70.90</td>
<td>71.18</td>
<td>71.39</td>
<td>71.24</td>
<td>70.83</td>
</tr>
<tr>
<td>Re-sampling</td>
<td>68.86</td>
<td>69.98</td>
<td>70.55</td>
<td>70.63</td>
<td>70.51</td>
<td>70.20</td>
</tr>
<tr>
<td>Regression</td>
<td>69.33</td>
<td>69.78</td>
<td>70.17</td>
<td>70.04</td>
<td>69.76</td>
<td>69.12</td>
</tr>
<tr>
<td>Selected features</td>
<td>69.24</td>
<td>70.08</td>
<td>70.46</td>
<td>70.39</td>
<td>70.10</td>
<td>69.76</td>
</tr>
</tbody>
</table>
Table 5.17. ROUGE-1 F-measure results (%) on test set for ASR outputs.

<table>
<thead>
<tr>
<th>compression ratio</th>
<th>13%</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline (all features)</td>
<td>59.77</td>
<td>60.76</td>
<td>61.85</td>
<td>62.39</td>
<td>62.79</td>
<td>63.23</td>
</tr>
<tr>
<td>all features</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Up-sampling</td>
<td>64.74</td>
<td>65.87</td>
<td>66.53</td>
<td>66.82</td>
<td>66.89</td>
<td>66.89</td>
</tr>
<tr>
<td>Down-sampling</td>
<td>63.89</td>
<td>64.79</td>
<td>65.05</td>
<td>65.30</td>
<td>65.28</td>
<td>65.18</td>
</tr>
<tr>
<td>Re-sampling</td>
<td>62.99</td>
<td>64.60</td>
<td>65.43</td>
<td>65.90</td>
<td>66.34</td>
<td>66.67</td>
</tr>
<tr>
<td>Regression</td>
<td>64.53</td>
<td>65.79</td>
<td>66.43</td>
<td>66.86</td>
<td>66.93</td>
<td>66.72</td>
</tr>
<tr>
<td>Selected features</td>
<td>63.73</td>
<td>64.84</td>
<td>65.69</td>
<td>66.05</td>
<td>66.21</td>
<td>66.12</td>
</tr>
</tbody>
</table>

5.4 Summary

In this chapter, we used a supervised learning approach for meeting summarization. Each sentence in the document was represented by a variety of features, capturing lexical, discourse, topic, and speaker information. A statistical classifier was used to decide whether this sentence should be included in the summary or not. We have thoroughly investigated three issues in this classification task, and our experimental results have shown significant performance improvement using our proposed methods.

Since the summary sentences (positive class in the task) are only a small percent of the entire document, we first studied the imbalanced data problem. We proposed three sampling approaches to address this problem, up-sampling, down-sampling and re-sampling. The idea of both up-sampling and down-sampling is to focus on those non-summary sentences that are most similar to the human annotated summaries, either changing their labels to positive or removing them from training. Re-sampling uses a different strategy, which selects sentences with high importance (based on similarity to the entire document) for both training and testing. All of these methods increase the positive to negative ratios in order to help the classifier better learn to discriminate the two classes. The best performance was achieved by different configurations (weighting meth-
ods and sampling rates), but overall, we have observed consistent performance gain using these sampling methods on both human transcripts and ASR conditions.

Second, we investigated using a regression model rather than binary classification for this task. Each training sentence was assigned a numerical weight according to its similarity to the annotated summary sentences. These new labels of the training samples represent fine-grained information and take into account human annotation disagreement. Our experimental results showed this regression model is quite robust and outperforms the binary classification setup.

Last, we used forward feature selection to evaluate the contribution of various features used for meeting summarization. We selected a subset of important features for human transcripts and ASR outputs respectively. Our analysis showed that the structural features are very important for both human transcripts and ASR outputs. The lexical features are more indicative for human transcripts, whereas for ASR outputs, the speaker related features are used more heavily. Our results have shown that using the subset of the features yielded better performance than using the entire feature set, especially for the ASR condition on the test set. However, when we used these selected features as the representation of each instance and combined with the sampling and regression approaches, there is no further improvement. We will investigate how to effectively combine these different approaches in our future work.
CHAPTER 6
FROM TEXT TO SPEECH SUMMARIZATION *

The methods we introduced in previous chapters can be applied to both text and speech summarization, since we do not use any speech specific information. In this chapter we investigate using speech information to improve the speech summarization performance, and addressing the problems caused by ASR errors.

Previous work has shown that combining acoustic/prosodic features with traditional textual features can further improve the summarization performance for some speech domains, such as broadcast news and lectures. Less analysis has been conducted for meeting summarization. So far most of meeting summarization research using supervised learning has focused on lexical and structural features, such as [32]. In this chapter, we will analyze the effectiveness of prosodic features, aiming to answer the question how they can be more effectively used for meeting summarization, and whether it is possible to construct a good text-independent summarizer by only using the prosodic features. Furthermore, since the acoustic and textual features can be considered as conditionally independent, we will investigate using co-training algorithm to increase the classification accuracy by leveraging the information from a large amount of unlabeled data.

In the experimental results shown in previous chapters, we find that the performance using the ASR output is consistently lower (to different extent) comparing to that using human transcripts no.

*© 2010 IEEE. Reprinted, with permission, from IEEE Transactions on Audio, Speech and Language Processing, Using N-best Lists and Confusion Networks for Meeting Summarization, Shasha Xie and Yang Liu
matter whether supervised or unsupervised approaches were used. To address the problem caused by imperfect recognition transcripts, in this chapter we will also investigate using rich speech recognition results for summarization.

6.1 Integrating Prosodic Features in Extractive Meeting Summarization †

Compared to text summarization that relies on lexical, syntactic, positional and structural information, speech summarization can leverage the additional sources of information contained in speech, such as speaker and acoustic/prosodic information. These provide important information for summarization as they represent how a sentence is said other than what it is said. Several recent studies have evaluated the impact of traditional textual features and speech-specific acoustic/prosodic features using classifier-based methods in speech summarization [12, 33, 35, 39, 72]. For the broadcast news domain, [33, 35] showed that the best performance was obtained by combining acoustic features with lexical, structural and discourse features; however, using only acoustic and structural features can achieve good performance when speech transcription is not available. Similar findings are also presented in [72] using acoustic and structural features for Mandarin broadcast news. In contrast, [12] showed different patterns for lecture summarization than broadcast news domain. The acoustic and structural features are less important due to the fact that the speaking styles of anchors and reporters are relatively consistent in broadcast news, whereas the speaking styles of lecture speakers vary a lot. In this section, we analyze the effectiveness of prosodic features, aiming to answer the question how they can be more effectively used for meeting summarization, and whether it is possible to construct a good text-independent summarizer by only using the prosodic features.

†Joint work with Dilek Hakkani-Tür and Benoit Favre
6.1.1 Acoustic/Prosodic Features

Following previous research on speech summarization, we first extract 13 original features using Praat [73]. We have five F0 related features representing the minimum, maximum, median, mean value of F0, and the range of F0 for each instance. Similarly, we extract five energy features also for the minimum, maximum, median, mean value of energy, and the range of energy of each sample. We include a duration feature which is the length of the sentence in seconds. Two speaking rate features are used, which are the sentence duration divided by the number of words and the number of letters in each sentence, respectively.

For these prosodic features, in addition to the raw values, we investigate different normalization methods based on various information.

- **Speaker-based normalization**
  Meeting recordings have multiple speakers. In general, speakers have different pitch, energy and speaking rates. In this normalization measure, each of the feature values is normalized using the mean and variance values of that feature for each speaker.

- **Topic-based normalization**
  Meetings can often be divided into several parts, each with its own topic based on the discussion. We assume that a speaker may have different prosodic behaviors for different topics according to his/her interest, roles, or conversation partners in a topic discussion. Therefore in this method, all the feature values are normalized using the mean and variance values for a topic. Note that this normalization is performed for each speaker.

- **Local window-based normalization**
This method does not rely on content information like the topic-based normalization or use only the information from the speaker himself. We expect that the speakers are affected by other participants and may adjust their speaking rates, pitch, or energy according to who they are talking to in a local context. We simply use the previous and the following $N$ instances to normalize the feature values.

Following the idea of local window normalization, we expect that the differences between the current sentence and its neighbors in terms of prosodic cues can indicate the importance of the current sentence, therefore we propose to include prosodic delta features — the difference between the current instance’s feature values to its previous $M$ and next $M$ instances. The idea of computing delta features has been widely used in tasks such as speech and speaker recognition to represent dynamic information.

6.1.2 Experiments

We first present experimental results for the development set to evaluate various factors, including prosodic feature normalization methods, effect of prosodic delta features, and combination of prosodic and non-prosodic features. Then we demonstrate the final results on the test set.

Baseline Results

The baseline in our experiments is using all the non-prosodic features we described in Section 5.1.1. Table 6.1 shows the ROUGE-1 (unigram match) F-measure scores for the human transcripts and ASR output. For the ASR condition, all the non-prosodic features are extracted from the ASR transcripts for both training and testing.
Table 6.1. ROUGE-1 F-measure results (%) of the baselines by selecting longest sentences and using non-prosodic features on development set.

<table>
<thead>
<tr>
<th>compression ratio</th>
<th>REF</th>
<th>13%</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>non-prosodic</td>
<td>67.25</td>
<td>67.80</td>
<td>67.76</td>
<td>67.56</td>
<td>67.22</td>
<td>66.86</td>
</tr>
<tr>
<td>ASR</td>
<td>non-prosodic</td>
<td>62.05</td>
<td>63.35</td>
<td>64.24</td>
<td>64.64</td>
<td><strong>64.77</strong></td>
<td>64.38</td>
</tr>
</tbody>
</table>

Results of Using Acoustic/Prosodic Features

Tables 6.2 and 6.3 show the ROUGE results for human transcripts and ASR output respectively when using only the acoustic/prosodic features described in Section 6.1.1. We present results using the raw values of the prosodic features, as well as adding different normalized features. For a comparison, results using non-prosodic features are also included in the tables.

We can see that using the raw prosodic features underperforms the baseline, with more difference on ASR output. It is worth pointing out that the prosodic features and the output confidence scores are the same for human transcripts and ASR output (they only rely on speech signals). When selecting summary sentences according to a predefined compression ratio, different transcripts (human vs. ASR output) are used to select the segments for these two conditions. The degraded performance on the ASR condition is mainly due to the high WER.

Results in Tables 6.2 and 6.3 show that in general there is a consistent improvement when using feature normalization. Adding speaker normalized prosodic features performs better than raw values on both human transcripts and ASR output, which is consistent with the findings in the domain of broadcast news summarization [35]. Adding topic normalization we can further improve the performance on human transcripts. Note that after speaker and topic normalization, the performance on human transcripts has already outperformed the baseline of using non-prosodic features.
Table 6.2. ROUGE-1 F-measure results (%) using raw values and different normalization methods of acoustic/prosodic features on development set for human transcripts.

<table>
<thead>
<tr>
<th>ratio</th>
<th>baseline</th>
<th>prosodic features</th>
<th>with normalization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>raw value</td>
<td>speaker</td>
</tr>
<tr>
<td>13%</td>
<td>67.25</td>
<td>65.74</td>
<td>65.75</td>
</tr>
<tr>
<td>14%</td>
<td><strong>67.80</strong></td>
<td>66.13</td>
<td>66.35</td>
</tr>
<tr>
<td>15%</td>
<td>67.76</td>
<td><strong>66.42</strong></td>
<td>66.84</td>
</tr>
<tr>
<td>16%</td>
<td>67.56</td>
<td>66.14</td>
<td>66.91</td>
</tr>
<tr>
<td>17%</td>
<td>67.22</td>
<td>65.78</td>
<td><strong>67.09</strong></td>
</tr>
<tr>
<td>18%</td>
<td>66.86</td>
<td>65.42</td>
<td>66.96</td>
</tr>
</tbody>
</table>

Table 6.3. ROUGE-1 F-measure results (%) using raw values and different normalization methods of acoustic/prosodic features on development set for ASR output.

<table>
<thead>
<tr>
<th>ratio</th>
<th>baseline</th>
<th>prosodic features</th>
<th>with normalization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>raw value</td>
<td>speaker</td>
</tr>
<tr>
<td>13%</td>
<td>62.05</td>
<td>59.54</td>
<td>60.07</td>
</tr>
<tr>
<td>14%</td>
<td>63.35</td>
<td>60.75</td>
<td>61.61</td>
</tr>
<tr>
<td>15%</td>
<td>64.24</td>
<td>61.59</td>
<td>62.55</td>
</tr>
<tr>
<td>16%</td>
<td>64.64</td>
<td>62.12</td>
<td>63.39</td>
</tr>
<tr>
<td>17%</td>
<td><strong>64.77</strong></td>
<td>62.37</td>
<td><strong>62.31</strong></td>
</tr>
<tr>
<td>18%</td>
<td>64.38</td>
<td><strong>62.56</strong></td>
<td><strong>63.84</strong></td>
</tr>
</tbody>
</table>
However, for ASR condition, adding topic information degrades the ROUGE scores. That’s probably because the speech recognition accuracy is different for each topic. The local window based normalization is the most effective one among these three normalization methods for both human transcripts and ASR output. We try different window length for human transcripts and ASR output respectively. The best window size for human transcripts is half of the document length, but for ASR output a smaller window is assigned (1/9 of the document size). This normalization method yields a significantly better score than the baseline (69.03 vs. 67.56) on human transcripts, but the difference on ASR output is much less.

Although using the raw values of prosodic features does not perform as well as using non-prosodic features, we can see that with proper normalization, we obtain better performance than the baseline using the non-prosodic features. This shows the feasibility of extracting the summary without text information. Because the worse performance on ASR output is mainly caused by the WER, the performance drop when using ASR output indicates that the selected summary sentences have recognition errors. [13] showed that summary sentences have lower WER than the average WER. But we can still see the WER has a great influence on summarization performance for the meeting domain.

The experimental results of adding the delta features are shown in Table 6.4 using the best normalization setup (local window normalization). The delta features are the difference between the current instance’s feature values and its previous and next M instances. We also try different M values, and the best one is 4 for human transcripts and 5 for ASR output. We notice that adding the prosodic differences substantially improves the performance, with more gain on human transcripts than ASR output. Comparing with the results presented in previous work using a large
set of features including lexical, structural, discourse and prosodic features [13, 34, 74], we obtain state-of-the-art results by only using the acoustic/prosodic features.

Table 6.4. ROUGE-1 F-measure results (%) of adding delta features on development set.

<table>
<thead>
<tr>
<th>Compression ratio</th>
<th>13%</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>REF</strong> window norm</td>
<td>67.65</td>
<td>68.25</td>
<td>69.01</td>
<td>69.03</td>
<td>68.63</td>
<td>68.35</td>
</tr>
<tr>
<td>+delta</td>
<td>69.9</td>
<td>70.8</td>
<td><strong>71.18</strong></td>
<td><strong>71.18</strong></td>
<td>70.69</td>
<td>70.49</td>
</tr>
<tr>
<td><strong>ASR</strong> window norm</td>
<td>62.51</td>
<td>63.58</td>
<td>64.46</td>
<td>65.02</td>
<td>65.62</td>
<td><strong>65.63</strong></td>
</tr>
<tr>
<td>+delta</td>
<td>63.42</td>
<td>64.65</td>
<td>65.78</td>
<td>66.07</td>
<td><strong>66.31</strong></td>
<td><strong>66.31</strong></td>
</tr>
</tbody>
</table>

To evaluate the effect of different prosodic features, we performed remove-one feature evaluation using human transcripts. In order to better understand the impact of the basic prosodic features, in this experiment, we only used the raw prosodic features and their local window normalized values. Furthermore, since the prosodic modeling part for human transcripts and ASR output is the same, we use results on the human transcripts to avoid the confounding effect of ASR errors in calculating the ROUGE scores. We list the five most and least effective features together with their performance loss in Table 6.5. The ranking of the features is obtained based on the performance change when removing the feature from the entire feature set.

Table 6.5. The five most and least effective prosodic features evaluated using human transcripts on development set.

<table>
<thead>
<tr>
<th>Most Effective Features</th>
<th>Less Effective Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy range (5.1%)</td>
<td>f0 median (-0.3%)</td>
</tr>
<tr>
<td>Normalized f0 median (4.7%)</td>
<td>normalized energy mean (0.7%)</td>
</tr>
<tr>
<td>energy mean (4.3%)</td>
<td>speaking rate (letter) (0.7%)</td>
</tr>
<tr>
<td>energy median (3.8%)</td>
<td>normalized energy maximum (1.0%)</td>
</tr>
<tr>
<td>f0 maximum (3.1%)</td>
<td>normalized f0 mean (1.1%)</td>
</tr>
</tbody>
</table>

Combination with Non-prosodic Features

Finally we investigate if these prosodic features combine well with the non-prosodic features to further improve the system performance. We use two different combination methods. First is the feature level combination. We combine the non-prosodic features with the prosodic feature set that yielded the best results (basic acoustic/prosodic features with local window normalization and delta features) in one large feature set. The second one is a decision level combination. We train separate models for these two information sources and then for each test instance linearly combine the confidence scores from the two models. The final summary is constructed by selecting the instances with higher combined confidence scores. The experimental results are presented in Table 6.6, along with the individual results using prosodic or non-prosodic information only. For the decision level combination method, we varied the combination weights for human transcripts and ASR output respectively, and show the best results here.

From the results, we can see that feature level combination hurts the summarization performance compared to using one information source only, and there is more degradation on human transcripts. However, we observe performance improvement using decision level combination for both human transcripts and ASR output. Interestingly, we notice that for human transcripts, a higher weight was given to the prosodic model (0.7 for prosodic and 0.3 for non-prosodic). This is consistent with the individual model performance — the results of prosodic features are much better than the non-prosodic ones. For ASR condition, equal weights (0.5 and 0.5) were used for the two models, which can be explained in part by the fact that the two systems have similar performance.
Table 6.6. ROUGE-1 F-measure results (%) of integrating prosodic and non-prosodic information, in comparison with using only one information source on development set.

<table>
<thead>
<tr>
<th>compression ratio</th>
<th>13%</th>
<th>14%</th>
<th>15%</th>
<th>16%</th>
<th>17%</th>
<th>18%</th>
</tr>
</thead>
<tbody>
<tr>
<td>REF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>non-prosodic</td>
<td>67.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>prosodic</td>
<td>69.9</td>
<td>70.8</td>
<td><strong>71.18</strong></td>
<td><strong>71.18</strong></td>
<td>70.69</td>
<td>70.49</td>
</tr>
<tr>
<td>feature combine</td>
<td>66.10</td>
<td><strong>66.70</strong></td>
<td>66.61</td>
<td>66.51</td>
<td>66.40</td>
<td>65.84</td>
</tr>
<tr>
<td>decision combine</td>
<td>70.50</td>
<td>70.92</td>
<td><strong>71.40</strong></td>
<td>70.91</td>
<td>70.67</td>
<td>70.18</td>
</tr>
<tr>
<td>ASR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>non-prosodic</td>
<td>62.05</td>
<td>63.35</td>
<td>64.24</td>
<td>64.64</td>
<td><strong>64.77</strong></td>
<td>64.38</td>
</tr>
<tr>
<td>prosodic</td>
<td>63.42</td>
<td>64.65</td>
<td>65.78</td>
<td>66.07</td>
<td><strong>66.31</strong></td>
<td><strong>66.31</strong></td>
</tr>
<tr>
<td>feature combine</td>
<td>63.06</td>
<td>64.16</td>
<td>64.75</td>
<td>65.40</td>
<td><strong>65.44</strong></td>
<td>65.23</td>
</tr>
<tr>
<td>decision combine</td>
<td>64.15</td>
<td>65.36</td>
<td>66.12</td>
<td>66.44</td>
<td><strong>67.10</strong></td>
<td>67.02</td>
</tr>
</tbody>
</table>

Results on Test Set

The results on the test set are provided in Table 6.7 for both human transcripts and ASR output, using only non-prosodic or prosodic information, and their combination. ROUGE-2 scores (bigram matching) are also included in order to provide more information for comparison. We selected the best setup based on the results on the development set and applied it to the test set. The prosodic feature set includes local window normalization and delta features. The combined system is based on a decision level combination of the prosodic and non-prosodic models. We observe similar trends as on the development set. Using only the prosodic features we obtain better performance than non-prosodic information, and the combination of the models yields further improvement. These results are consistent across human transcripts and ASR output, ROUGE-1 and ROUGE-2 scores. We also verified that the results are significantly better than the baseline according to a paired t-test ($p < 0.05$).
Table 6.7. ROUGE-1 F-measure results (%) on test set.

<table>
<thead>
<tr>
<th>compression ratio</th>
<th>ROUGE-1 F-measure</th>
<th>ROUGE-2 F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13%</td>
<td>14%</td>
</tr>
<tr>
<td>REF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>non-prosodic</td>
<td>68.29</td>
<td>69.15</td>
</tr>
<tr>
<td>prosodic</td>
<td>68.87</td>
<td>69.85</td>
</tr>
<tr>
<td>combined</td>
<td>69.64</td>
<td>70.54</td>
</tr>
<tr>
<td>ASR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>non-prosodic</td>
<td>59.77</td>
<td>60.76</td>
</tr>
<tr>
<td>prosodic</td>
<td>64.55</td>
<td>65.18</td>
</tr>
<tr>
<td>combined</td>
<td>65.14</td>
<td>65.91</td>
</tr>
</tbody>
</table>

6.2 Semi-Supervised Extractive Speech Summarization via Co-Training Algorithm

Although the experimental results in Chapter 4 and 5 showed that supervised learning approaches achieved better performance than unsupervised ones, learning a summarization classifier requires a large amount of labeled data for training. Summary annotation is often difficult, expensive, and time consuming. Annotation of meeting recordings is especially hard because the documents to be summarized are transcripts of natural meetings that have very spontaneous style, contain many disfluencies, have multiple speakers, and are less coherent in content, and probably have a lot of errors if the transcripts are from automatic speech recognition system. It is very hard to read and understand the document, not to mention extracting the summary. On the contrary, meeting recordings and their transcripts are relatively much easier to collect. This situation creates a good opportunity for semi-supervised learning that can use large amount of unlabeled data, together with the labeled data, to build better classifiers. This technique has been shown to be very promising in many speech and language processing tasks, such as question classification, web classification, word-sense disambiguation and prosodic event detection [75, 76, 77, 78]. In this section,
we use the co-training algorithm [79] to explore semi-surprised learning in speech summarization. Co-training assumes that features can be split into two sets, which are conditionally independent given the class, and each of which is sufficient to train a good classifier. Unlike in text summarization task (where only textual information is available), we can easily extract two different views of features for speech summarization: one from the textual transcripts, and the other from the speech recordings.

6.2.1 Co-Training Algorithm for Meeting Summarization

Co-training algorithm was introduced to increase the classification accuracy by exploiting the information from a large amount of unlabeled data, together with a small set of labeled data [79]. Co-training assumes that the features can be split into two independent sets, and each set is sufficient to train a good classifier. Initially two separate classifiers are trained with the labeled data, on the two sub-feature sets respectively. Each classifier then classifies the unlabeled data, selects the samples that they feel most confident with, and uses these automatically labeled samples along with the original labeled data to “teach” the other classifier. This process iterates until the classification performance stabilizes, or all the unlabeled data is used, or after certain number of iterations. There are several possible ways to apply co-training algorithm to extractive speech summarization.

We investigate two methods in this study.

\[\text{§Note that here we use textual features to represent the feature set extracted from non-prosodic information, which includes speaker and topic information other than plain texts.}\]
**Sentence-Based Selection**

In the classification setup for extractive speech summarization, each training or testing instance is a sentence from the document to be summarized, and positive or negative labels are used to indicate whether or not this sentence is in the summary. We therefore use sentence as the basic selection unit in each co-training iteration. Precisely, $p$ unlabeled sentences are labeled as positive (summary sentences) and $n$ unlabeled sentences are labeled as negative in each iteration based on the confidence scores of current classifier’s prediction. These $p + n$ sentences are then added into the original training set to form a new train set. The detail of the algorithm is described in Algorithm 2, which basically follows the standard procedure of co-training.

Note that for extractive summarization task, the positive samples are only a small percent of all the instances in the document (i.e. summary is always compact). For example, in the ICSI meeting corpus the average percentage of the positive samples is 6.62%. In order to be consistent with the original training data distribution, we select more negative samples in each iteration than positive ones. We use $n = \alpha p$, where $\alpha$ is the ratio of the number of negative samples to the number of positive samples in the corpus ($\alpha = 15$ in our case).

**Document-Based Selection**

In sentence-based selection, the classifier labels sentences independently, regardless which document a sentence belongs to. In summary annotation, however, the decision for each sentence is not made independently, rather it is made by considering the entire document. This is a key difference between the classification setup for summarization vs. other classical classification tasks. In order to be consistent with human labeling process, we could use document as the basic selection
Algorithm 2 Co-Training for Extractive Speech Summarization

Let \( L \) be the set of labeled training sentences. Let \( U \) be the set of unlabeled training sentences. Each sentence is represented by two feature sets \( \{F_1, F_2\} \), representing textual and prosodic features respectively.

while \( U \neq \emptyset \) do
    Train the first classifier \( C_1 \) on \( L \) using \( F_1 \).
    Train the second classifier \( C_2 \) on \( L \) using \( F_2 \).
    for each classifier \( C_i (i = 1, 2) \) do
        (a) For each sentence in \( U \) (represented by \( F_i \)), \( C_i \) predicts its posterior probabilities of being labeled as positive;
        (b) \( C_i \) chooses \( p \) sentences (\( P \)) that it most confidently labels as positive and \( n \) sentences (\( N \)) that it most confidently labels as negative from \( U \);
        (c) \( C_i \) removes \( P \) and \( N \) from \( U \);
        (d) \( C_i \) adds \( P \) to \( L \) with positive labels, and \( N \) to \( L \) with negative labels.
    end for
end while

unit such that the entire document can be included into the training set. Similar to the sentence-based selection method above, we select the documents that the classifiers are most confident with, and adding all the sentences in these documents into the training set for next iteration. To assign a confidence score to each document, we take the classifier’s average confidence scores for all the sentences in the document. For each sentence in the document, we estimate the classifier’s confidence by taking the negative entropy of the posterior distribution. The confidence score of a document \( D \) is then measured as the average negative entropy of all the sentences in \( D \). At each iteration, the documents with high confidence scores are selected into the labeled training set. For each of the selected documents, we select the top \( \frac{100}{1+\alpha} \% \) of its sentences that the classifier most confidently labels as positive. These sentences are labeled as positive, and the rest as negative.
6.2.2 Experimental Results

For co-training, we first train two classifiers using the labeled data, on the textual and prosodic feature sets respectively. Then the additional training samples are iteratively selected according to the predictions from each classifier. After co-training, we have two classifiers trained from the textual and prosodic features respectively. We then extract the summaries for each document in the development set using these two classifiers. In order to evaluate the effect of the size of the initial labeled data, we use different size, starting from 10% of the training data to all of it. When a subset of the training set is used, the rest is treated as unlabeled data (since we do not have additional meeting data with similar style and topics). The subset of labeled data is randomly selected from the training set. For each setup, we run 10 independent trials, and report the average score. Figure 6.1 shows the co-training results using textual features (upper figure) and prosodic features (lower graph) on the development set. For a comparison, we also include the baseline results of supervised learning (dotted line in the figure) that just uses the initial labeled data without any unlabeled data, and is trained using all the features.

From Figure 6.1, we can see that for the supervised baseline, in general performance improves when using more data for training, which is as expected. The worst results are always observed when the labeled training data is 10% of the original data set. However, the best result is not obtained using all the labeled data for training. The reason might be that more labeled data probably will induce more noise for training, because we only have one annotator for the documents in the training set, and previous work showed that the human annotation agreement for summarization task is really low. We also run the experiments of only using textual or prosodic features for training. And the results using all features are better than those only using textual features, but worse
than using prosodic features. Overall, the baseline summarization results are still very competitive comparing to those reported in previous work.

Figure 6.1. ROUGE-1 F-measure results of co-training on the development using the models trained from textual and prosodic features.

Using sentence-based selection, co-training achieves significantly better performances than that of only using the labeled data, on both textual and prosodic features. Consistent improvements are achieved on different sizes of labeled data. The improvement is more obvious when the labeled training data set is small. When increasing the labeled data set, the differences between the co-training and baseline results decreases, since the unlabeled data that co-training algorithm exploits is getting smaller. Another interesting observation is that our preliminary experiments show that the performance using prosodic features are consistently better than those using textural features, but the results on textual features are remarkably improved by co-training, which are now competitive to the results of using prosodic features. This may be partly because of the original better performance of the prosodic classifier that adds more correct samples for textural classifier training.
For the document-based sample selection, there is no consistent improvement over the baseline results, and for some of the setups it is even worse than the baseline. One possible reason for that is that during co-training, although we select the documents with the highest confidence scores at each iteration, the classifier is not necessarily confident to all of the sentences in the document. In other words, the chance of adding misclassified samples increases, which could have potential impact on the learning and labeling process in the following iterations and thus hurt the co-training performance. Nevertheless, we believe the selection criteria can be improved by making it more directly optimized for the summarization task. Possible future directions include making a partial document selection, i.e., only adding the most confident samples in the selected document such that the selected samples can be both coherent and confident. From another point of view, since the supervised summarization classification approach itself is only dependent on individual sentences, and does not use any document level decision, it does not need document information for its iterative training process (features based on document information have been prepared offline already). This is different from active learning where the selection units have to be documents for humans to create reference labels.

The test set results are shown in Figure 6.2 using textual features (upper figure) and prosodic features (lower). Similar to the results on the development set, co-training algorithm with sentence-based selection can effectively use the information of unlabeled data to improve the summarization performance. The improvement is more significant when the labeled data set is small. Again, no consistent improvements are obtained using documents as the selection units.
Figure 6.2. ROUGE-1 F-measure results of co-training results on the test using the models trained from textual (upper) and prosodic features (lower).

6.3 Using N-best Lists and Confusion Networks for Meeting Summarization

Rich speech recognition results, such as N-best hypotheses and confusion networks, were first used in multi-pass ASR systems to improve speech recognition performance [52, 80]. They have been widely used in many subsequent spoken language processing tasks, such as machine translation, spoken document retrieval, and information extraction. For machine translation, [81] used n-best list for re-ranking by optimizing the interpolation weights for ASR and translation. In [82], the authors used confusion networks for machine translation, but they did not observe much improvement over n-best hypotheses. Lattices have been studied intensively for spoken document retrieval and indexing [83, 84], with reported better performance than just using 1-best ASR output. [85] used n-best lists for named entity recognition in speech data. [86] investigated using confusion networks for name entity detection and extraction and user intent classification. Both obtained better performance than using ASR 1-best output. There are also studies trying to more
tightly couple ASR and language processing components. For example, [87] proposed a joint decoding approach for speech translation. However, these systems are often too complex and hard to optimize, and do not always outperform those using loose coupling (i.e., using a pipeline approach, where ASR is following by subsequent language processing modules, but with n-best lists, lattices, or confusion networks as the interface in between). There is very limited research using more than 1-best ASR output for speech summarization. In [19, 22], the authors introduced acoustic confidence scores of the best hypotheses in order to minimize the WER in the summaries. They found that this can also help improve the summarization performance. Similar findings were presented in [88] – using the word confidence scores can significantly improve the summarization accuracy. In [89], confusion networks and position specific posterior lattices were considered in a generative summarization framework for Chinese broadcast news summarization, yielding better summarization results. In this section, we study the feasibility of using these information for meeting summarization task on the framework of MMR.

6.3.1 Rich Speech Recognition Output

In our study, we consider two different structures of rich speech recognition results, n-best hypotheses and confusion networks. We expect that the rich information contained in them (i.e., more word candidates and associated posterior probabilities) can help us better determine the importance of words and sentences for summarization.
N-best Hypotheses

The output of a speech recognition system is typically the best-matching sequence of words with the highest sentence posterior probability given the input speech signal. In addition to outputting 1-best hypotheses, the system can generate N-best candidates for each sentence.\(^\dagger\) Table 6.8 shows an example of 5-best hypotheses along with their confidence scores.\(\|^\|\) Often these sentences are very similar, but sometimes the difference is very crucial, which may have a great impact on summarization results. In this example, the word “breasts” in the top hypothesis is a substitution error. The third hypothesis contains the correct word “breath”.

Table 6.8. An example of 5-best hypotheses for a sentence segment.

<table>
<thead>
<tr>
<th>n-best hypotheses</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 if it’s picking up breasts noise</td>
<td>0.873</td>
</tr>
<tr>
<td>2 if it’s picking up breast noise</td>
<td>0.810</td>
</tr>
<tr>
<td>3 if it’s picking up breath noise</td>
<td>0.793</td>
</tr>
<tr>
<td>4 if it’s picking up breaths noise</td>
<td>0.788</td>
</tr>
<tr>
<td>5 if it’s picking up press noise</td>
<td>0.785</td>
</tr>
</tbody>
</table>

Confusion Networks

Confusion networks (CNs) have been used in many natural language processing tasks. Figure 6.3 shows the corresponding CN for the example we used for n-best hypotheses in Table 6.8. It

\(^\dagger\)Note that we are using “sentences” loosely here to represent the segments from ASR systems. These are generally different from linguistic sentences.

\(\|^\|\)These scores are different from the typical scores associated with n-best hypotheses from ASR output. The latter uses the combined acoustic and language model scores computed during decoding directly, whereas here we generate these probability scores from the confusion networks. See Section 6.3.3 for more information.
is a directed word graph from the starting node to the end node. Each edge represents a word with its associated posterior probability. There are several word candidates for each position. “-” in the CN represents a NULL hypothesis. Each path in the graph is a sentence hypothesis, and the best one consists of words with the highest probabilities for each position. For this example, “if it’s picking up breasts noise” is the best hypothesis, the same as that shown in Table 6.8. Compared to n-best lists, confusion networks are a more compact and powerful representation containing much more word candidates and sentence hypotheses.

Figure 6.3. An example of confusion networks for a sentence segment.

6.3.2 Extractive Summarization Using Rich ASR Output

The summarization framework we use is the unsupervised approach MMR introduced in Section 2.1.1 and 4.1.1. Since the input for summarization is rich speech recognition results, which are different from texts we used before, in this section we describe how MMR is adapted to n-best lists and confusion network input.
Using N-best Hypotheses for Summarization

We can easily extend the MMR approach to consider the n-best hypotheses for each sentence segment. The task now is to determine the segments, as well as the hypotheses, to be selected into the summary. Suppose for a sentence segment, there are N hypotheses \{h_{i1} \cdots h_{iN}\}. For each hypothesis \(h_{ij}\), its score can be calculated as follows:

\[
MMR(h_{ij}) = \lambda \times Sim_1(h_{ij}, D) - (1 - \lambda) \times Sim_2(h_{ij}, Summ)
\]  

(6.1)

where \(D\) is the vector for the entire document, which is formulated using the 1-best hypothesis of each segment. All the hypotheses in one segment are treated separately, that is, we calculate the salience score for each hypothesis, as shown in the formula above. In the iterative MMR selection, once a hypothesis is selected into the summary, the other hypotheses for that segment will not be considered for the following process. The final summary is composed of all the selected hypotheses.

For the similarity measure between two text segments, we can still use cosine similarity (Equation 4.1) and TFIDF term weighting for each word. Since confidence scores (or posterior probabilities) are available for sentence hypotheses, we propose to use that information to compute IDF for a word. In our method, we use a word’s soft occurrence in a meeting document, which is defined as the maximum posterior probability among all the hypotheses (for all the segments of a meeting) that contain this word. The IDF of a word is then calculated as:

\[
IDF(w_i) = \log\left(\frac{N}{\sum_{D_j}(\text{max}_p p_j(w_i))}\right)
\]  

(6.2)

where \(\text{max}_p p_j(w_i)\) represents the soft occurrence of word \(w_i\) in document \(D_j\). We add these soft counts for all the documents to obtain the denominator in Equation 6.2. Different from the tradi-
tional IDF calculation, where the number of documents containing a word is an integer number, here the denominator is a real number.

Another modification we make to MMR is to use the confidence score associated with each sentence hypothesis to compute its salience score, that is,

$$MMR(h_{ij}) = \lambda \times Sim_1(h_{ij}, D) \times p(h_{ij}) - (1 - \lambda) \times Sim_2(h_{ij}, Summ)$$

(6.3)

where $p(h_{ij})$ is the confidence score of hypothesis $h_{ij}$. This allows the system to select the hypotheses which not only are representative, but also have higher recognition confidence.

The MMR method is a greedy algorithm and selects each summary segment iteratively. When we expand the candidate summary sentences from 1-best to n-best hypotheses, the time complexity increases significantly. The more candidates we consider, the slower the system will be. One approximation we used in Section 4.1.1 is to consider only a subset of sentences rather than all the sentences in the document. This subset of candidate sentences is selected based on their similarity score to the entire document (i.e., the first term in the MMR formula). We found that this did not degrade performance, but can significantly speed up the extraction process. For n-best lists, we also adopt this approximation to pre-select some segment candidates for summarization. The score for each segment is estimated using the highest score (similarity with the entire document) among all the hypotheses for that segment.

**Using Confusion Networks for Summarization**

To use the confusion networks under the MMR framework, we need to re-create the vectors for each summarization unit $S_i$, the entire document $D$, and the current selected summary $Summ$ in
Equation 2.1. The vectors for $D$ and $Summ$ are formed by simply concatenating the corresponding segments $S_i$ together. We consider two ways to construct the vector for each segment. The first method is simply using all the word candidates in the CNs without considering any confidence measure or posterior probability information. Then the same TFIDF term weighting method can be used for each word as before, i.e., counting the number of times a word appears (TF) and how many documents in which it appears (used to calculate IDF). The second method leverages the confidence scores to build the vectors. For TF of a word $w_i$, we calculate it by summing up its posterior probabilities $p(w_{ik})$ at each position $k$ in the segment, that is,

$$TF(w_i) = \sum_k p(w_{ik})$$

(6.4)

IDF is calculated similar to Equation 6.2, where the soft occurrence $max_j p_j$ is a word’s maximum probability among all the positions it occurs in the CNs for a document $D_j$.

The vectors for the text segments can be constructed using the entire confusion networks, or the pruned one, in which the words with lower posterior probabilities are removed beforehand. This can avoid the impact of noisy words, and increase the system speed as well. We investigate three different pruning methods, listed below.

- **absolute pruning**: In this method, we delete words if their posterior probabilities are lower than a predefined threshold, i.e., $p(w_i) < \theta$.

- **max_diff pruning**: First for each position $k$, we find the maximum probability among all the word candidates: $P_{max_k} = \max_j p(w_{jk})$. Then we remove a word $w_i$ in this position if the absolute difference of its probability with the maximum score is larger than a predefined threshold, i.e., $P_{max_k} - p(w_{ik}) > \theta$. 

• max_ratio pruning: This is similar to the above one, but instead of absolute difference, we use the ratio of their probabilities, i.e., \( \frac{p(w_{ik})}{P_{max_k}} < \theta \).

Again, for the last two pruning methods, the comparison is done for each position in the CNs.

**Summary Rendering**

There are two different ways to provide the final summary to the users, using a textual summary or speech segments. If the input for summarization is human transcripts or 1-best ASR output, there is only one hypothesis for each segment, and the final textual summary can be naturally constructed using the selected sentences from MMR. In the case of using n-best lists, the summarization process can determine which segments and which hypotheses for those segments (there can be only one hypothesis selected for a segment) to include in the summary. Therefore after the MMR process, we can obtain a textual summary that is simply the concatenation of the selected hypotheses. However, when using the confusion networks as the representation for each segment, we only know which segments are selected by the summarizer. To create a textual summary, we can use the best recognition hypotheses from CNs for those selected segments.

The other summary output is in the form of speech segments. We can return the selected speech segments to the users to allow them to play them back. To measure the system performance using this kind of output, we use the corresponding reference transcripts of the selected segments to construct the final textual summary in order to apply text-based evaluation metrics. This will allow us to focus on the evaluation of the correctness of the selected summary segments.
6.3.3 Experimental Results and Analysis

The Generation of Rich Speech Recognition Results

We obtained the n-best list (including 1-best results) for each segment from the corresponding confusion networks. The confidence score for each utterance hypothesis is the average of all the words’ posterior probabilities used to form the sentence. The extractive summarization units we use are the sentence segments from the ASR system. This is often different from the syntactic or semantic meaningful units (e.g., sentences), but is a more realistic setup. Most of the previous studies for speech summarization used human labeled sentences as extraction units (for human transcripts, or map them to ASR output), which is not the real scenario when performing speech summarization on the ASR condition. In the future, we will use automatic sentence segmentation results, which we expect are better units than pause-based segmentation used in ASR.

Results Using 1-best Hypothesis and Human Transcripts

Our first set of experiments is performed using the development set. We generate the baseline summarization results using 1-best ASR output. The MMR method as described in Section 4.1 is used, and the term weight for each word is the traditional TFIDF value. For a comparison, we also perform summarization using the corresponding human transcripts for each ASR segment. Note that this is still different from using human transcripts and human labeled DA segments (because of the ASR segments). We generated summaries with the word compression ratio ranging from 13% to 18%, and only provide the best results in this section. The ROUGE-1 and ROUGE-2 results are presented in Table 6.9. Comparing the results for the two conditions, ASR output and human transcripts, we can see the performance degradation due to recognition errors. The difference
between them seems to be large enough to warrant investigation of using rich ASR output for improved summarization performance.

Table 6.9. ROUGE F-measureresults (%) using 1-best hypotheses and human transcripts on the development set. The summarization units are the ASR segments.

<table>
<thead>
<tr>
<th></th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline: 1-best</td>
<td>65.60</td>
<td>26.83</td>
</tr>
<tr>
<td>Human transcript</td>
<td>69.98</td>
<td>33.21</td>
</tr>
</tbody>
</table>

**Summarization Results Using N-best Hypotheses**

In our study, we consider up to 50-best hypotheses for summarization. The oracle WER using 50-best hypotheses is 36.50% for the 6 meetings in the development set, compared to 41.51% using 1-best hypotheses. The other metric we define for quality measure of n-best lists is the word coverage, which is the percentage of the words (measured using word types) in human transcripts that appear in the n-best hypotheses. We consider this metric since we use a vector space model under the MMR summarization framework. If a correct word appears in a hypothesis, it will be included in the segment's vector no matter whether it appears in the correct position or not. This is a more lenient measure than WER. The word coverage using 50-best hypotheses is 82.65% on the development set, compared to 71.55% for 1-best output.

We compare three methods using n-best lists for summarization.

- We use the traditional way to compute the IDF values without using any confidence scores of the hypotheses. This simply considers all the hypotheses in n-best lists. This method is referred as ‘Nbest’. 
• We calculate the IDF values using the confidence scores as shown in Equation 6.2. We call this method ‘Nbest-IDF’.

• In addition to using the confidence for IDF calculation, we also use the probability of a hypothesis to calculate its salience score in MMR (Equation 6.3). This method is called ‘Nbest-IDF-SIM1’.

Figure 6.4. ROUGE-1 F-measure results using n-best lists (up to 50 hypotheses) on the development set.

The summarization results for different numbers of hypotheses are presented in Figure 6.4 and 6.5 using ROUGE-1 and ROUGE-2 scores respectively. The dotted horizontal lines show the baseline results when using 1-best hypotheses without considering any confidence information. For both ROUGE-1 and ROUGE-2 results, we can see consistent improvement using the proposed methods compared to the original 1-best baseline. There is incremental improvement when more and more information is used – more hypotheses, then confidence scores for IDF calcula-
Figure 6.5. ROUGE-2 F-measure results using n-best lists (up to 50 hypotheses) on the development set.

tion, and finally confidence scores for sentence salience measure during MMR selection. For the three setups, the best ROUGE-1 results (peak points in the curves) are obtained by using different numbers of hypotheses. Among all the configurations, the best ROUGE-1 score is 0.6806, which is obtained by considering the top 10 hypotheses, and using the confidence scores for IDF calculation and MMR iteration. For ROUGE-2 score, the improvements by considering more hypotheses and their confidence scores are more obvious. The best score is 0.2970, obtained by using the top 30 hypotheses, and confidence scores for calculating IDF values and sentence significance scores.

There is still a gap between these best results using n-best lists and those using human transcripts shown in Table 6.9. We also noticed that a large percentage of the selected summary hypotheses are not the original top hypotheses on the n-best lists. For instance, for the best setup obtaining the highest ROUGE-1 scores, 59.7% of its selected hypotheses are the non-best hypotheses, indicating
that information from the additional hypotheses is helpful for summarization.

In order to investigate whether we select the correct sentence segments, next we provide the results of using the corresponding human transcripts for the selected segments to generate the final summary. This can avoid the impact of WER and allow us to focus on the correctness of selected segments. Results are shown in Figure 6.6 and 6.7. We also include the results using 1-best hypotheses and human transcripts (Ref) for a comparison. For the 1-best case, we use the corresponding human transcripts to form the summary once the segment selection process is done. The result for the human transcripts condition is the same as that in Table 6.9.

![Figure 6.6. ROUGE-1 F-measure results using n-best lists (up to 50 hypotheses) on the development set, with the summaries constructed using the corresponding human transcripts of the selected hypotheses.](image)

Comparing with the results in Figure 6.4 and 6.5, the ROUGE scores are much higher when we concatenate the human transcripts of the selected segments to form the final summary, which shows that the word errors in the ASR output still impact the summary quality. We also notice
Figure 6.7. ROUGE-2 F-measure results using n-best list (up to 50 hypotheses) on the development set, with the summaries constructed using the corresponding human transcripts of the selected hypotheses.

that the difference between using 1-best hypothesis and human transcripts is greatly reduced using this new summary formulation. This suggests that the incorrect word hypotheses do not have a very negative impact in terms of selecting summary segments. Our proposed methods of using the n-best hypotheses and their confidence scores outperform the baseline results using 1-best hypotheses. In Figure 6.6, for some setups we observe better ROUGE-1 scores than using the human transcripts, for example, when the number of hypotheses is 25 in Nbest-IDF-SIM1. This superior performance over using human transcripts is more obvious based on ROUGE-2 scores (Figure 6.7). For most of the setups, we obtain better performance than using human transcripts. These results show that if we only consider the task of selecting the most representative and correct segments, using n-best hypotheses can compensate the performance degradation due to imperfect ASR output. However, if we need to present textual summaries using recognition output, we
will still be penalized using the selected hypotheses, although it is much better than the summary extracted using 1-best hypotheses.

**Summarization Results Using Confusion Networks**

**A. Characteristics of CNs**

First we perform some analysis of the confusion networks using the development set data. We define two measurements:

- **Word coverage.** This is to verify that CNs contain more correct words than the 1-best hypotheses. It is defined as the percentage of the words in human transcripts (measured using word types) that appear in the CNs. We use word types in this measurement since we are using a vector space model and the multiple occurrence of a word only affects its term weights, not the dimension of the vector. Note that for this analysis, we do not perform alignment that is needed in word error rate measure — we do not care whether a word appears in the exact location; as long as a word appears in the segment, its effect on the vector space model is the same (since it is a bag-of-words model).

- **Average node density.** This is the average number of candidate words for each position in the confusion networks.

Figure 6.8 shows the analysis results for these two metrics, which are the average values on the development set. In this analysis we used absolute pruning method, and the results are presented for different pruning thresholds. For a comparison, we also include the results using the 1-best hypotheses (shown as the dotted line in the figure), which has an average node density of 1, and
the word coverage of 71.55%. When the pruning threshold is 0, the results correspond to the original CNs without pruning.

![Graph](image)

Figure 6.8. Average node density and word coverage of the confusion networks on the development set.

We can see that the confusion networks include much more correct words than 1-best hypotheses (word coverage is 89.3% vs. 71.55%). When increasing the pruning thresholds, the word coverage decreases following roughly a linear pattern. When the pruning threshold is 0.45, the word coverage of the pruned CNs is 71.15%, lower than 1-best hypotheses. For node density, the non-pruned CNs have an average density of 11.04. With a very small pruning threshold of 0.01, the density decreases rapidly to 2.11. The density falls less than 2 when the threshold is 0.02, which means that for some nodes there is only one word candidate preserved after pruning (i.e., only one word has a posterior probability higher than 0.02). When the threshold increases to 0.4, the density is less than 1 (0.99), showing that on average there is less than one candidate left for
each position. This is consistent with the word coverage results — when the pruning threshold is larger than 0.45, the confusion networks have less word coverage than 1-best hypotheses because even the top word hypotheses are deleted. Therefore, for our following experiments we only use the thresholds $\theta \leq 0.45$ for absolute pruning.

Note that the results in the figure are based on absolute pruning. We also performed analysis using the other two pruning methods described in Section 6.3.2. For those methods, because the decision is made by comparing each word’s posterior probability with the maximum score for that position, we can guarantee that at least the best word candidate is included in the pruned CNs. We varied the pruning threshold from 0 to 0.95 for these pruning methods, and observed similar patterns as in absolute pruning for the word coverage and node density analysis. As expected, the fewer word candidates are pruned, the better word coverage and higher node density the pruned CNs have.

**B. Effect of segmentation representation**

We evaluate the summarization performance of different vector representations using confusion networks. Table 6.10 shows the results on the development set using various input under the MMR framework. The results using 1-best and human transcripts are included in the table for a comparison. The third row in the table, best hyp (wp), uses the 1-best hypothesis, but the term weight for each word is calculated by considering its posterior probability in the CNs (denoted by “wp”). We calculate the TF and IDF values using the methods proposed in Section 6.3.2. The other representations in the table are for the non-pruned and pruned CNs, and with or without using the posteriors to calculate term weights.

In general, we find that using confusion networks improves the summarization performance
Table 6.10. ROUGE F-measure results (%) on the development set using different vector representations based on confusion networks: non-pruned and pruned, using posterior probabilities (“wp”) and without using them.

<table>
<thead>
<tr>
<th>segment representation</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best hyp</td>
<td>65.60</td>
<td>26.83</td>
</tr>
<tr>
<td>Best hyp (wp)</td>
<td>66.83</td>
<td>29.84</td>
</tr>
<tr>
<td>Non-pruned CNs</td>
<td>66.58</td>
<td>28.22</td>
</tr>
<tr>
<td>Non-pruned CNs (wp)</td>
<td>66.47</td>
<td>29.27</td>
</tr>
<tr>
<td>Pruned CNs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Absolute</td>
<td>67.44</td>
<td>29.02</td>
</tr>
<tr>
<td>Absolute (wp)</td>
<td>66.98</td>
<td>29.99</td>
</tr>
<tr>
<td>Max_diff</td>
<td>67.29</td>
<td>28.97</td>
</tr>
<tr>
<td>Max_diff (wp)</td>
<td>67.10</td>
<td>29.76</td>
</tr>
<tr>
<td>Max_ratio</td>
<td>67.43</td>
<td>28.97</td>
</tr>
<tr>
<td>Max_ratio (wp)</td>
<td>67.06</td>
<td>29.90</td>
</tr>
<tr>
<td>Human transcript</td>
<td>69.98</td>
<td>33.21</td>
</tr>
</tbody>
</table>

upon the baseline. Since CNs contain more candidate words and posterior probabilities, a natural question to ask is, which factor contributes more to the improved performance? We can compare the results in Table 6.10 across different conditions that use the same candidate words, one with standard TFIDF, and the other with posteriors for TFIDF; or that use different candidate words and the same setup for TFIDF calculation. Our results show that there is more improvement using our proposed method of considering posterior probabilities for TFIDF calculation, especially for ROUGE-2 scores. Even when just using 1-best hypotheses, if we consider posteriors, we can obtain very competitive results. There is also a difference in the effect of using posterior probabilities across different inputs. When using the top hypotheses representation, posteriors help both ROUGE-1 and ROUGE-2 scores; when using confusion networks, non-pruned or pruned, using posterior probabilities improves ROUGE-2 results, but not ROUGE-1. Our results show that adding more candidates in the vector representation does not necessarily help summarization. Using the pruned CNs yields better performance than the non-pruned ones. There is not much
Table 6.11. ROUGE F-measure results (%) on the development set using different segment representations, with the summaries constructed using the corresponding human transcripts for the selected segments.

<table>
<thead>
<tr>
<th>segment representation</th>
<th>ROUGE-1</th>
<th>ROUGE-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best hyp</td>
<td>68.26</td>
<td>32.25</td>
</tr>
<tr>
<td>Best hyp (wp)</td>
<td>69.16</td>
<td>33.99</td>
</tr>
<tr>
<td>Non-pruned CNs</td>
<td>69.28</td>
<td>33.49</td>
</tr>
<tr>
<td>Non-pruned CNs (wp)</td>
<td>67.84</td>
<td>32.95</td>
</tr>
<tr>
<td>Max diff</td>
<td>69.88</td>
<td>34.17</td>
</tr>
<tr>
<td>Max_diff (wp)</td>
<td>69.38</td>
<td>33.94</td>
</tr>
<tr>
<td>Max ratio</td>
<td>69.76</td>
<td>34.06</td>
</tr>
<tr>
<td>Max_ratio (wp)</td>
<td>69.44</td>
<td>34.39</td>
</tr>
<tr>
<td>Human transcript</td>
<td>69.98</td>
<td>33.21</td>
</tr>
</tbody>
</table>

difference among different pruning methods. Overall, the best results are achieved by using pruned CNs: best ROUGE-1 result without using posterior probabilities, and best ROUGE-2 scores when using posteriors.

C. Presenting summaries using human transcripts

Table 6.11 shows the results of confusion networks where the summaries are constructed using the corresponding human transcripts of the selected segments. We can see that the summaries are much better comparing with the results presented in Table 6.10. Using the best hypotheses with their posterior probabilities we can obtain similar ROUGE-1 score and slightly higher ROUGE-2 score comparing to the results using human transcripts. There is a further performance improvement using the pruned CNs.

Note that when using the non-pruned CNs and posterior probabilities for term weighting, the ROUGE scores are worse than most of other conditions. We performed some analysis and found
that one reason for this is the selection of some poor segments. Most of the word candidates in the non-pruned CNs have very low confidence scores, resulting in high IDF values using our proposed methods. Since some top hypotheses are NULL words in the poorly selected summary segments, it did not affect the results when using the best hypothesis for evaluation. But when using human transcripts, it leads to lower precision and worse overall F-scores. This is not a problem for the pruned CNs since words with low probabilities have been pruned beforehand, and thus do not impact segment selection. We will investigate better methods for term weighting to address this issue in our future work.

These experimental results prove that using more word candidates and their confidence scores can provide better vector representations in the MMR method and help select the correct sentence segments. The high word error rate does not significantly impact the process of selecting summary segments, and using CNs we can achieve similar or sometimes even slightly better performance than using human transcripts.

**Word Error Rate of Extracted Summaries**

Some previous work showed that using word confidence scores helped minimize the WER of the extracted summaries, which then led to better summarization performance [19, 22]. In [46], the authors also showed that the summarization systems tend to select utterances with lower ASR errors. In this section, we perform some analysis on the relationship between WER and summarization performance for different summarization approaches.

Based on the results in Section 6.3.3, we selected the summarization outputs obtained from different setups for n-best lists and confusion networks that yielded the best ROUGE-1 and ROUGE-
Table 6.12. WER (%) of extracted summaries using n-best hypotheses and confusion networks on the development set.

<table>
<thead>
<tr>
<th></th>
<th>WER1</th>
<th>WER2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (Best hyp)</td>
<td>33.11</td>
<td>33.65</td>
</tr>
<tr>
<td>N-Best</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nbest</td>
<td>33.16</td>
<td>31.59</td>
</tr>
<tr>
<td>Nbest-IDF</td>
<td>32.89</td>
<td>31.51</td>
</tr>
<tr>
<td>Nbest-IDF-SIM1</td>
<td>31.52</td>
<td>30.73</td>
</tr>
<tr>
<td>CNs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best hyp (wp)</td>
<td>33.80</td>
<td>30.96</td>
</tr>
<tr>
<td>Non-pruned</td>
<td>31.90</td>
<td>30.69</td>
</tr>
<tr>
<td>Non-pruned (wp)</td>
<td>36.62</td>
<td>35.47</td>
</tr>
<tr>
<td>Pruned</td>
<td>33.77</td>
<td>30.85</td>
</tr>
<tr>
<td>Pruned (wp)</td>
<td>33.48</td>
<td>30.63</td>
</tr>
<tr>
<td>Entire document</td>
<td>41.51</td>
<td></td>
</tr>
</tbody>
</table>

2 scores respectively. We calculate the WER by comparing the system generated summaries to the corresponding human transcripts of the selected summary segments. The WER results are presented in Table 6.12, where WER1 is obtained using the summaries with the best ROUGE-1 scores, and WER2 from the summaries yielding the best ROUGE-2 scores. Note that the sentences used for WER calculation are generated by different summarization systems and are thus different for different rows in the table. In the last row, we also include the WER for the entire development set.

It is obvious that the extracted summaries have lower WER for all the conditions, compared to the WER for the entire set. This is consistent with some previous findings [46], suggesting that to some extent the summarization system tends to select utterances with lower ASR errors. Between the two WERs, the summaries with better ROUGE-2 scores have lower WER than those with better ROUGE-1 scores. For n-best lists, we observe consistent WER reduction with more information considered in the summarization system, which also results in improved summarization performance in general (see Figure 6.4). Because the selected sentence segments are different
for different setups, we can not simply conclude that the improvement of summarization performance is from the reduction of WER. For example, there is 60.7% overlap between the summary segments extracted by the baseline and the “Nbest-IDF-SIM1” system. For these overlapped segments, we observe a WER reduction from 28.85% for the 1-best baseline system to 28.40% using the “Nbest-IDF-SIM1” system, suggesting that some of the non-top hypotheses selected by the “Nbest-IDF-SIM1” system as summary sentences also have lower WER.

For confusion networks, there is no consistent pattern between WER and summarization performance (comparing the results in Table 6.12 and 6.10). In particular, using non-pruned confusion networks yields the worst summarization performance, but the best WER results. When using confusion networks, the summarization system selects the most representative segments, but can not determine the correct words in a segment to use in the final summary. This is different from using n-best hypotheses, where the system evaluates each hypothesis individually and has a measurement of their goodness during summarization process. Overall we can see that lower WER in the summary does not guarantee better summarization performance. Even though the baseline system (using 1-best hypothesis) has the worst summarization performance among all the setups, the WER of the baseline is not the highest.

Although we find that for some of the cases the WER of the selected summaries is lower than the baseline, we believe the main reason for the improved summarization performance in our study is because of the selection of better segments, as demonstrated in Figure 6.6 and 6.7 for n-best hypotheses, and Table 6.11 for confusion networks. When we use the corresponding human transcripts for the selected segments to present the summary, we obtain similar or even better ROUGE scores than using human transcripts.
Comparison of Using N-Best Hypotheses and Confusion Networks

So far we have demonstrated improved summarization performance from both n-best lists and confusion networks, evaluated using both ROUGE-1 and ROUGE-2 scores. In addition, if the task is to select summary segments and present them to the users, or equivalently, if the evaluation is done using the corresponding human transcripts for the selected segments, both of them can achieve competitive or even better performance comparing with using human transcripts as the input of the summarization system.

Between n-best lists and CNs, there are some differences that are worth pointing out. First is related to the computational complexity using the MMR method for these two structures. For n-best hypotheses, the running time of the system increases roughly linearly with the number of hypotheses we use for each segment.*** For confusion networks, we construct the vector for each segment using all the word candidates in it. As we mentioned earlier, confusion networks are a more compact representation than n-best lists. The vector size for CNs is much less than \( n \) times of that for 1-best hypotheses (\( n \) corresponds to \( n \)-best lists). If we use pruned CNs, then the increase is even less. The average node density can be reduced to just slightly over one, yet preserving a good tradeoff between word coverage and the size of the CN [90]. Therefore the running time of MMR does not increase as much as in n-best condition, and is not significantly different from using 1-best hypothesis.

The second point is regarding the WER of the extracted summaries. As shown in Section 6.3.3, **There is some subtle difference compared to increasing the document length by \( n \) times. Since we use a constraint in MMR that none of the hypotheses for a segment will be considered once a hypothesis for that segment is selected into the summary, in the iterative process, we are not considering all the hypotheses.**
using n-best hypotheses the system is able to select the “best” hypotheses to construct the final summary. These are often not the original 1-best hypotheses from the ASR system, and they result in better WER in the extracted summaries. For confusion networks, we still use the top hypotheses to form the summary after the segments are selected. There is no obvious WER reduction in the summaries extracted using CNs, and the summarization performance improvements are mainly from better selection of the summary segments.

Note that the n-best hypotheses we use in this study are generated from the confusion networks, different from what is typically done in ASR systems. This is mainly to guarantee that we use similar information for n-best lists and CNs, such as the word candidates, sentence hypotheses, and their posterior probabilities, so that we can more fairly compare results using these two different structures. However, we expect that most of the findings in this study are not specific to the particular n-best lists we used.

**Experimental Results on Test Set**

The summarization results on the test set are presented in Table 6.13, using the baseline, n-best hypotheses, confusion networks, and human transcripts. For each condition, the final summary is evaluated using the best hypotheses or the corresponding human transcripts of the selected segments. The summarization system setups (e.g., \( \lambda \) value in the MMR formula, word compression ratio) used for the test set are decided based on the results on the development set.

The trends for the test set are similar to the development set. Using rich speech recognition results can efficiently improve the summarization performance evaluated by both ROUGE-1 and ROUGE-2 scores. For n-best lists, the best results are obtained by considering more sentence
Table 6.13. ROUGE F-measure results (%) on the test set using different inputs for summarization, and different forms of summaries (using recognition hypotheses or human transcripts of the selected summary segments).

<table>
<thead>
<tr>
<th>Segment Representation</th>
<th>Best Hypotheses</th>
<th>Human Transcripts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-1</td>
<td>R-2</td>
</tr>
<tr>
<td>Baseline (Best hyp)</td>
<td>65.73</td>
<td>26.79</td>
</tr>
<tr>
<td>N-best</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nbest</td>
<td>66.10</td>
<td>27.22</td>
</tr>
<tr>
<td>Nbest-IDF</td>
<td>66.19</td>
<td>27.29</td>
</tr>
<tr>
<td>Nbest-IDF-SIM1</td>
<td><strong>66.87</strong></td>
<td><strong>27.83</strong></td>
</tr>
<tr>
<td>CNs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best hyp (wp)</td>
<td>65.92</td>
<td>27.27</td>
</tr>
<tr>
<td>Pruned CNs</td>
<td><strong>66.47</strong></td>
<td><strong>27.73</strong></td>
</tr>
<tr>
<td>Human transcript</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

hypotheses and the sentence posterior probabilities during IDF calculation and MMR iteration.

The improvement is statistically significant comparing to the baseline results according to a paired t-test ($p < 0.01$). For confusion networks, the results are also significantly better than the baseline system ($p < 0.05$). If the summaries are constructed using the corresponding human transcripts of the extracted segments in order to evaluate the correctness of the selected segments, for both rich speech recognition structures we observe similar or even better results than those obtained using human transcripts (the difference is not statistically significant). This shows that using rich speech recognition results can compensate for the impact from recognition errors and still allow us to select correct summary segments.

### 6.4 Summary

Compared to text summarization, more information can be exploited for meeting summarization, such as speaker and acoustic/prosodic information. In this chapter, we evaluated how we can effectively use the speech specific information to help improving summarization performance.
First, we investigated the way to represent acoustic/prosodic features, and to integrate them with traditional textual features for meeting summarization. For prosodic information, we adopted three normalization methods based on different sources of information: speaker, topic, and local context information. Our experimental results showed that these normalization methods improve the performance compared with only using the raw values. We also demonstrated additional gain when adding the delta features to represent how a sentence is different from its neighbors. When using only the prosodic features, we were able to outperform the baseline of using the non-prosodic features. In addition, we evaluated different approaches to integrate prosodic and non-prosodic information, and showed that a decision level combination can improve summarization performance upon that of the individual models.

Summary annotation for speech recordings is time-consuming and expensive. We adopt the semi-supervised learning to increase the classification accuracy by leveraging the information from a large amount of unlabeled data. Co-training is known to be very effective if the features can be divided into two conditionally independent sets. For speech summarization, two different sets of features, textual features and acoustic/prosodic features, are naturally available, making co-training a good choice of semi-supervised speech summarization. Our experimental results showed that co-training algorithm with sentences as the selection units can effectively improve the summarization performance. Both classifiers trained using textual and prosodic features work better than only using the labeled data, with more gain for the relatively weak classifier (using textual features).

We observed performance degradation when using ASR output for meeting summarization mainly because of word errors. To address this problem, we proposed to use rich speech recognition results for speech summarization under an unsupervised MMR framework. We evaluated two
different structures, n-best hypotheses and confusion networks. Using n-best hypotheses, we pro-
posed to treat each hypothesis separately, and use the confidence scores of the hypotheses for IDF
calculation and MMR selection. For confusion networks, we introduced a vector representation
for the segments by considering more word candidates and their associated posterior probabili-
ties. Our experimental results showed that using rich speech recognition results can significantly
improve the summarization performance. The gain is from the use of confidence scores, as well
as more candidate words. We also demonstrated that it is possible to use rich speech recognition
results to achieve similar or even better performance than using human transcripts if the goal is to
select the right segments. This is important since one possible rendering of summarization results
is to return the audio segments to the users, which does not suffer from recognition errors.

Our analysis showed that the WER of the extracted summaries is lower than that on the entire
document. However, we did not find consistent relationship between lower WER and better sum-
marization performance. Using n-best hypotheses allows us to select the best sentence hypotheses
for summarization, which may be different from the originally top ranked recognition hypotheses,
and yield better recognition accuracy for the extracted summaries. However, the system complexity
using n-best list is roughly linearly increased compared to using 1-best hypotheses. Using confu-
sion networks can help select better segments for summarization, but does not provide information
to re-rank or select word candidates. It does have a running time advantage since the vector size
is quite small, similar to that using 1-best hypotheses. Therefore, the choice of which rich speech
recognition results to use depends on factors such as the speed requirement of the summarization
system and the readability and the users’ requirements of the extracted summary.
CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 Conclusion

The goal of this thesis is to extract the most relevant sentences from a meeting to form its summary. Meeting summarization is more challenging compared to summarization of written text and other speech genres because of the presence of disfluencies, multiple participants and often high speech recognition error rate. We addressed several issues in existing unsupervised and supervised text summarization approaches, and showed significant system improvement on both human transcripts and ASR output. We also analyzed the speech specific information, such as extracting acoustic/prosodic features for summarization, and using rich speech recognition results to improve the performance on ASR output.

Unsupervised approaches are relatively simple and do not need any labeled data for training. The first unsupervised method we investigated is maximum marginal relevance (MMR), where the important sentences are selected iteratively according to each sentence’s similarity to the entire document and its similarity to the already extracted summary. We compared three different similarity measures: cosine similarity, centroid score, and corpus-based semantic similarity. Our experiments showed that the last two approaches achieved better performance than cosine similarity, because they can better capture the semantic level information other than simple lexical matching. Another unsupervised method we evaluated is a concept-based global optimization framework.
Under this framework, *concepts* are defined and used as the minimum units for summarization, and a subset of summary sentences are extracted to cover as many concepts as possible. However, one problem of this method is that it tends to select shorter sentences in order to increase the concept coverage, which results in the degradation of the linguistic quality of selected summaries. We proposed leveraging sentence level information, measured using the cosine similarity of each sentence to the entire document, to extract the summaries that can cover both important concepts and sentences. We evaluated different ways to use sentence level information, and demonstrated better summarization performance with the sentence level information comparing to the original framework.

Extractive summarization can be considered as a binary classification problem, where each sentence is represented by a large set of features, and positive or negative label is assigned to indicate whether the sentence is in the summary or not. We analyzed three issues using supervised learning models, the imbalanced data problem, the human annotation disagreement, and feature effectiveness. First, we proposed different sampling methods to deal with the imbalanced data problem, where the summary sentences are the minority class. After re-labeling or re-selecting the training samples, we can provide more balanced data for training a classifier, and therefore we observed significantly better summarization performance. Second, in order to account for human disagreement for summary annotation, we reframed the extractive summarization task using a regression scheme instead of binary classification. Each sample in the training set was assigned a numerical label according to its importance, and we proved that this fine-grained annotation can help us train a better classifier. Last, we evaluated the contribution of different features used for meeting summarization using forward feature selection. We showed that our proposed topic-
related features were important for selecting the summary sentences, and using a subset of selected features outperformed using all the features.

The speech recordings are available for meeting summarization, so that we can use more speech specific information to help improve the summarization performance. In supervised learning framework, we incorporated normalized acoustic/prosodic features, and investigated how they can be combined with the textual information. We showed that by only using prosodic features we can obtain comparable or even better performance than using the textual features. Further improvement can be obtained by combining these two sets of features at the decision level.

Since the features used for meeting summarization can come from two different sources: textual and speech information, and using each source alone can construct a good summarizer, we proposed using a co-training algorithm to improve the summarization performance by leveraging the information from a large amount of unlabeled data. Our experimental results showed that co-training with sentences as the selection units can effectively improve the performance for both classifiers trained using textual and prosodic features.

We observed consistent performance degradation when using the ASR output for summarization comparing to the results obtained using human transcripts. In order to improve the performance on the ASR condition, we proposed using rich speech recognition results. Two kinds of structures were considered, n-best hypotheses and confusion networks. Each of them contains much more word candidates and sentence hypotheses than 1-best ASR output. Under the unsupervised MMR framework, we developed new term weighting methods and text segment representations to utilize the rich information. We demonstrated that using rich speech recognition results can significantly improve the summarization performance. Moreover, if the task is to generate speech
summaries or identify salient segments, using rich speech recognition results can extract similar or even better summary segments than using human transcripts.

7.2 Future Work

There are many important direct extensions of our current work and future research directions for extractive meeting summarization. A few future work that are most relevant to the research in this thesis are listed below.

7.2.1 Automatic Sentence Segmentation

As we mentioned in Chapter 1, the output of ASR systems is only word sequence without punctuation marks or sentence segments. For extractive summarization, this ASR output needs to be first segmented into proper units (e.g., sentences). If we want to process this output, we first need to segment this sequence to sentences. In this thesis, for most of the experiments we used human annotated dialog acts as the summarization units on human transcripts or aligned them to ASR output. For the experiments in Section 6.3.1, we used the pause-based segments provided by the automatic speech recognizer. In the future, we will investigate using automatic sentence segmentation results, which we expect are better units than pause-based segmentation, but probably do not perform as well as human annotated segments. In our previous work, we used an HMM for sentence segmentation, and evaluated the effect of different segments on an MMR-based summarization system [16]. We showed that using the automatic segmentation degraded the summarization performance comparing to using human labeled segments. We will do more analysis of the impact of automatic sentence segments on the summarization performance under other summa-
rization frameworks, such as the supervised learning approaches. We will also evaluate whether we can improve the summarization performance using rich speech recognition results by adopting automatically segmented sentences.

7.2.2 Automatic Topic Segmentation

In Chapter 5, the textual features we introduced for supervised learning include topic-related information. In feature analysis we showed these topic-related features were selected as the important ones using both human transcripts and ASR output. The topic segments we used in this thesis are also human annotated. If we want to develop an automatic speech summarization system, we need to perform automatic topic segmentation. In our future work, we will investigate the impact of using automatic topic segments for speech summarization, and the effectiveness of the topic-related features extracted accordingly.

7.2.3 Semi-Supervised Learning for Meeting Summarization

In Section 6.2.1, we adopted the co-training algorithm to leverage unlabeled data. Semi-supervised learning methods are very useful techniques for meeting summarization since the annotation of meeting recordings is especially hard because of the spontaneous style of meeting transcripts. In the future work we will continue investigating other sample selection criteria that are more geared towards the summarization task. We will also investigate other semi-supervised learning algorithms, such as transductive SVM or graph-based semi-supervised learning approaches.
7.2.4 Using Rich Speech Recognition Results

In Section 6.3.1, two kinds of rich speech recognition structures were considered to improve the summarization performance on the ASR condition, n-best hypotheses and confusion networks. The MMR method was used as the basic summarization framework. We will investigate using n-best hypotheses or confusion networks under other summarization framework. For example, using the supervised learning approaches, we need to modify features that are used for human transcripts or 1-best hypotheses conditions, such as sentence length, cue words and phrases, and centroid scores. We can also include additional information about confidence scores in the feature set.
REFERENCES


[84] Tee Kiah Chia, Khe Chai Sim, Haizhou Li, and Hwee Tou Ng, “A lattice-based approach to query-by-example spoken document retrieval,” in Proc. of SIGIR, 2008.


VITA

I received my B.S. degree in electronic and information engineering from Tianjin University, Tianjin, China, in 2003, and the M.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2006. From August 2006, I have been pursuing the Ph.D. degree majoring in computer science at the University of Texas at Dallas.

My research interests include spoken language processing and understanding, extractive text/speech summarization, and information retrieval and extraction.

Publications:


on Acoustics, Speech and Signal Processing (ICASSP), Dallas, March 2010.


