CONTENT SELECTION IN MULTI-DOCUMENT SUMMARIZATION

Kai Hong

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2015

Supervisor of Dissertation	Co-Supervisor of Dissertation
____________________________	____________________________
Ani Nenkova, Associate Professor,	Mitchell P. Marcus, Professor,
Computer and Information Science	Computer and Information Science

Graduate Group Chairperson

____

Lyle Ungar, Professor, Computer and Information Science

Dissertation Committee:

John M. Conroy, IDA Center for Computing Sciences

Sampath Kannan, Professor, Computer and Information Science

Mark Liberman, Professor, Linguistics

Lyle Ungar, Professor, Computer and Information Science

CONTENT SELETION IN MULTI-DOCUMENT SUMMARIZATION

To Yumeng, without whom I would never have completed this dissertation.

Acknowledgments

First, I would like to thank my advisors, Ani Nenkova and Mitch Marcus. Ani gave me innumerable feedbacks on formulating research problems, designing experiments, and interpreting results. Her detailed suggestions on how to write papers and give talks also shape me into as a better researcher. I owe a debt of gratitude to Mitch, who has advised me throughout my dissertation stage. His invaluable comments greatly improved the quality of the thesis, as well as my understanding of computational linguistics. Furthermore, whenever I encountered difficulties, he was always there to help. I will never forget his encouragement, guidance, integrity, and unwavering support.

I would like to express my gratitude to John Conroy for being a wonderful collaborator and teaching me about proportion test and Philadelphia pretzel; to Sampath Kannan and Mark Liberman for insightful questions that lead me to look into problems from different perspectives; to Lyle Ungar for inspiring comments on research throughout my years at Penn.

Chris Callison-Burch served on my WPE-II committee and gave me detailed feedback. I thank him for his encouragement and steadfast belief in my ability. I am lucky to be surrounded by many outstanding colleagues at the Penn NLP group: Houwei Cao, Anne Cocos, Mingkun Gao, Junyi Li, Constantine Lignos, Xi Lin, Annie Louis, Ellie Pavlick, Emily Pitler, Daniel Preotiuc, Andy Schwartz, Joao Sedoc, Wei Xu, Rui Yan, and Qiuye Zhao. Thanks also go to Mike Felker, Cheyrl Hickey, and other administration staff at CIS.

I am indebted to many other people on my way of becoming a computer scientist. Eric Chang and Yan Xu were my mentors when I was doing internship at Microsoft Research Asia. I thank them for introducing me into natural language processing and collaborating on my first paper. I also enjoyed an unforgettable summer at Microsoft Research Silicon Valley, where my mentor, Dilek Hakkani-Tur, showed me what research is like in industry. I would also like to thank my other collaborators: Benoit Favre, Christian Kohler, Alex Kulesza, Hui Lin, Mary March, Amber Parker, Pengjun Pei, Junichi Tsujii, Ragini Verma, and Ye-Yi Wang.

I would also like to thank my friends for their encouragements and companion. An incomplete list would be: Chen Chen, Xue Chen, Arthur Azevedo De Amorim, Chang Guo, Yu Hu, Shahin Jabbari, Chaoran Li, Wei Li, Yang Li, Sheng Mao, Salar Moarref, Hua Qiang, Mukund Raghothaman, Chen Sun, Xiaofan Tong, Fanxi Wang, Zhirui Wang, Dafeng Xu, Meng Xu, Hongbo Zhang, Mabel Zhang, and Nan Zheng.

Finally, I would like to thank Yumeng Ou for being my significant other, encouraging me through the most difficult periods during graduate school. I have been more than fortunate to share with her all the happiness and sorrow through these years, and look forward to the wonderful years that lie ahead of us. My deepest gratitude goes to my parents, Xiang Hong and Shuang Liu, who cultivated my interest of knowledge, nurtured my curiosity, taught me the values of determination, diligence, and integrity, and supported me no matter what happens. Without their unconditional love, I would never have become who I am today.

ABSTRACT

CONTENT SELECTION IN MULTI-DOCUMENT SUMMARIZATION

Kai Hong

Ani Nenkova

Mitchell P. Marcus

Automatic summarization has advanced greatly in the past few decades. However, there remains a huge gap between the content quality of human and machine summaries. There is also a large disparity between the performance of current systems and that of the best possible automatic systems. In this thesis, we explore how the content quality of machine summaries can be improved.

First, we introduce a supervised model to predict the importance of words in the input sets, based on a rich set of features. Our model is superior to prior methods in identifying words used in human summaries (i.e., summary keywords). We show that a modular extractive summarizer using the estimates of word importance can generate summaries comparable to the state-of-the-art systems. Among the features we propose, we highlight global knowledge, which estimate word importance based on information independent of the input. In particular, we explore two kinds of global knowledge: (1) important categories mined from dictionaries, and (2) intrinsic importance of words. We show that global knowledge is very useful in identifying summary keywords that have low frequency in the input.

Second, we present a new framework of system combination for multi-document summarization. This is motivated by our observation that different systems generate very different summaries. For each input set, we generate candidate summaries by combining whole sentences produced by different systems. We show that the oracle summary among these candidates is much better than the output from the systems that we have combined. We then introduce a support vector regression model to select among these candidates. The features we employ in this model capture the informativeness of a summary based on the input documents, the outputs of different systems, and global knowledge. Our model achieves considerable improvement over the systems that we have combined while generating summaries up to a certain length. Furthermore, we study what factors could affect the success of system combination. Experiments show that it is important for the systems combined to have a similar performance.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Most summaries we encounter are manually written. However, as an unprecedented volume of textual data is generated every day, it is impossible for human to summarize all of them. Therefore, there exists a strong need for automatic summarization systems to be developed. Though research in this direction has evolved greatly in the past few decades, there remains a huge gap between the qualities of manually written summaries and machine summaries.[1]

There are different reasons that cause the gap in quality. Although people may think linguistic quality is the main reason, in fact human summaries also tend to cover much more content than machine summaries do. Hence, content selection remains a challenging problem in summarization. Indeed, according to the evaluation results from the Text Analysis Conference (TAC)[2] organized by NIST, all human summaries include much more informative content than the best machine summaries (Dang and Owczarzak, 2008) in terms of the Pyramid score (Nenkova et al., 2007), a semantically driven manually generated score that weighs a summary based on its content coverage.

[1] According to Rankel et al., (2011), assessors in the 2008 Text Analysis Conference (TAC) have commented that they can easily recognize which summaries are human or machine generated, even if the origin of summary is hidden.

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

To more concretely demonstrate the content gap between human and machine summaries, we show the summaries from a topic-based summarization problem in Table 1.1. The task is to produce a 100 word summary based on a set of input documents[3] that addresses a given topic. Compared to the human summary, the machine summary fails to effectively summarize the main topic and goes into too much details. Moreover, it fails to address the topic statement. The significant gap of content quality is also reflected by the Pyramid score, where the human summary scores three times as high as the machine summary.

Topic: Indian Pakistan conflict
Narrative/Topic statement: Describe efforts made toward peace in the India-Pakistan conflict over Kashmir.

Human summary
The Pyramid score = 0.720

There appears to be progress in the Indian-Pakistani dispute over Kashmir. The bilateral ceasefire has held for a year. The Indian Prime Minister met with Kashmiri political leaders for the first time. India has reduced the number of border troops and announced a multibillion-dollar development plan. Bus service has been opened for the first time in 57 years. The Pakistani President has suggested three solutions: Some areas of Kashmir could be made independent or placed under joint Indian-Pakistani control or put under UN administration. However, India has ruled out division along religious lines or redrawing international borders.

Machine summary
The Pyramid score = 0.235

CHAKOTHI, Pakistan Sixty-year-old Khalid Hussain hugged the brother-in-law he had only seen before in photos after traveling from Indian- to Pakistani-controlled Kashmir on the bus service revived between the divided region after a half-century of conflict. “We are so happy we can’t express how we feel,” Hussain said Thursday after crossing a narrow bridge over a rushing river that spans the military Line of Control that cleaves the disputed Himalayan region. Minutes later he embraced his wife’s brother, Asif Solaria, who lives on the Pakistan side and came to meet him.

Table 1.1: A human summary and a machine summary towards a topic-based summarization problem. The Pyramid score measures the coverage of informative content in the summary.

[3] Three input documents of this problem are provided in Appendix A.

The gap in quality is also huge between summaries generated by current systems and oracle automatic systems that use information from human gold-standard summaries. Many prior papers show this fact based on extractive summarization systems that select *sentences* from the input. Lin and Hovy (2003b) show that the potential of extractive based systems is high for single-document summarization. With the goal of quantifying an upper bound for extractive multi-document summarization (MDS), Conroy et al. (2006a) estimate word weights based on the probability distribution of words in human summaries. The authors then present a system to generate oracle summaries, which score much higher than automatic systems and even higher than human abstractors[4] evaluated by ROUGE (Lin, 2004), the de-facto automatic evaluation method. Gillick et al. (2009) provide another upper bound for extractive MDS by optimizing the coverage of n-grams in human summaries. They show that the best possible ROUGE-2 (which measures bigram overlap) far exceeds that of the state-of-the-art systems on the TAC 2009 data (0.20 vs 0.12).

Apart from automatic evaluation, the gap in quality is also large in terms of manual evaluation. This is confirmed by our experiments in Chapter 6, where we study the oracle performance of system combination for MDS. The oracle performance is derived by selecting the summary with the highest ROUGE-2 by exhaustive search. Even if we only combine four 100 word summaries (which includes about 14 unique sentences), the advantage of oracle over the current best systems (SumCombine, ICSISumm) is already large based on the Pyramid Method (see Table 1.2). The advantage of the oracle system will be larger if all input sentences are used.

From above, we can see that content quality of the summaries generated by

[4] Note that this does not necessarily mean human summaries are inferior to the oracle machine summaries, due to the following reasons. First, ROUGE serves as a surrogate for manual evaluations. Having higher ROUGE score does not necessarily indicate that one system has a higher manual evaluation score. Second, current manual evaluation methods can also be improved. It is possible that human summaries still have better content quality compared to the oracle machine summaries, which cannot be reflected by current manual evaluation methods. Third, even if the oracle summaries may indeed have a better content quality compared to human abstracts, they are likely to suffer from poor linguistic quality (e.g., coherence, grammaticality, coreference).

	Oracle	SumCombine	ICSISumm	Greedy-KL
The Pyramid score	0.626	0.549	0.530	0.459

Table 1.2: The Pyramid scores for Oracle (i.e., selects the candidate summary with the highest ROUGE-2) and three competitive baseline systems for the system combination task on the TAC 2008 dataset.

current systems is much inferior to that of human summaries. It is also much lower than the quality of the best possible summaries that can be generated by automatic systems. These facts motivate us to study how the content quality of machine summaries can be improved, which is the focus of this thesis.

Before introducing our work, we first present the high-level architecture of a summarization system. Knowing this is helpful to understand how the topics discussed in the thesis are related to improving the content quality of a summary. The summarization pipeline we present here is similar to the ones in Mani and Maybury (1999) and Mani (2001a). Specifically, the pipeline includes five stages: preprocessing, input analysis, transformation, generation, and synthesis.

Preprocessing: Sentence segmentation and word tokenization are performed at this stage. Moreover, as some information in the original documents are noisy or uninformative, some systems remove such information from the input sentences during preprocessing (Conroy et al., 2006a; Gillick et al., 2009).

Input analysis: This stage analyzes the documents to be summarized (input). Normally, summarization systems would first represent the input using *elements* at different levels of granularity: words, phrases, sentences, paragraphs and documents. Some systems also build connections between these elements (Erkan and Radev, 2004; Mihalcea, 2005; Wan and Yang, 2008). After that, summarizers estimate importance of these elements using different approaches. We call all the information above a representation of the input.

The input documents can be represented in different ways. For example, some systems build a table that includes words and the importance of these words, then estimate sentence importance based on word importance (Nenkova and Vanderwende, 2005; Conroy et al., 2006a; Yih et al., 2007). Topic based approaches represent the input and documents in the input into different semantic topics (Daumé III and Marcu, 2006; Haghighi and Vanderwende, 2009; Celikyilmaz and Hakkani-Tur, 2010).

Note that most prior work analyzes the input simply based on the documents to be summarized. In this thesis, we propose global knowledge, which aims to provide information that is independent of a particular input.

Transformation: This stage transforms input representations into summary representations. For example, systems that conduct sentence compression remove irrelevant information from input sentences to form summary sentences. Extractive summarization systems that use input sentences verbatim would simply skip this step (Radev et al., 2004b; Erkan and Radev, 2004; Nenkova and Vanderwende, 2005).

Generation: At this stage, a summarizer selects the content (e.g., words, sentences) to be included in the final summaries. Importance of the content (e.g., words, sentences) and diversity between these contents need to be considered.

Synthesis: At this stage, summarizers might refine or reorganize the selected sentences. For instance, sentence reordering can be performed in order to make the summaries more coherent. Coreference resolution can be included in order to make the summaries more readable. This stage is also optional.

Note that the boundaries between different summarization stages are sometimes obscure. For example, some systems optimize the content quality and coherence simultaneously, which combines the generation and synthesis stages (Christensen et

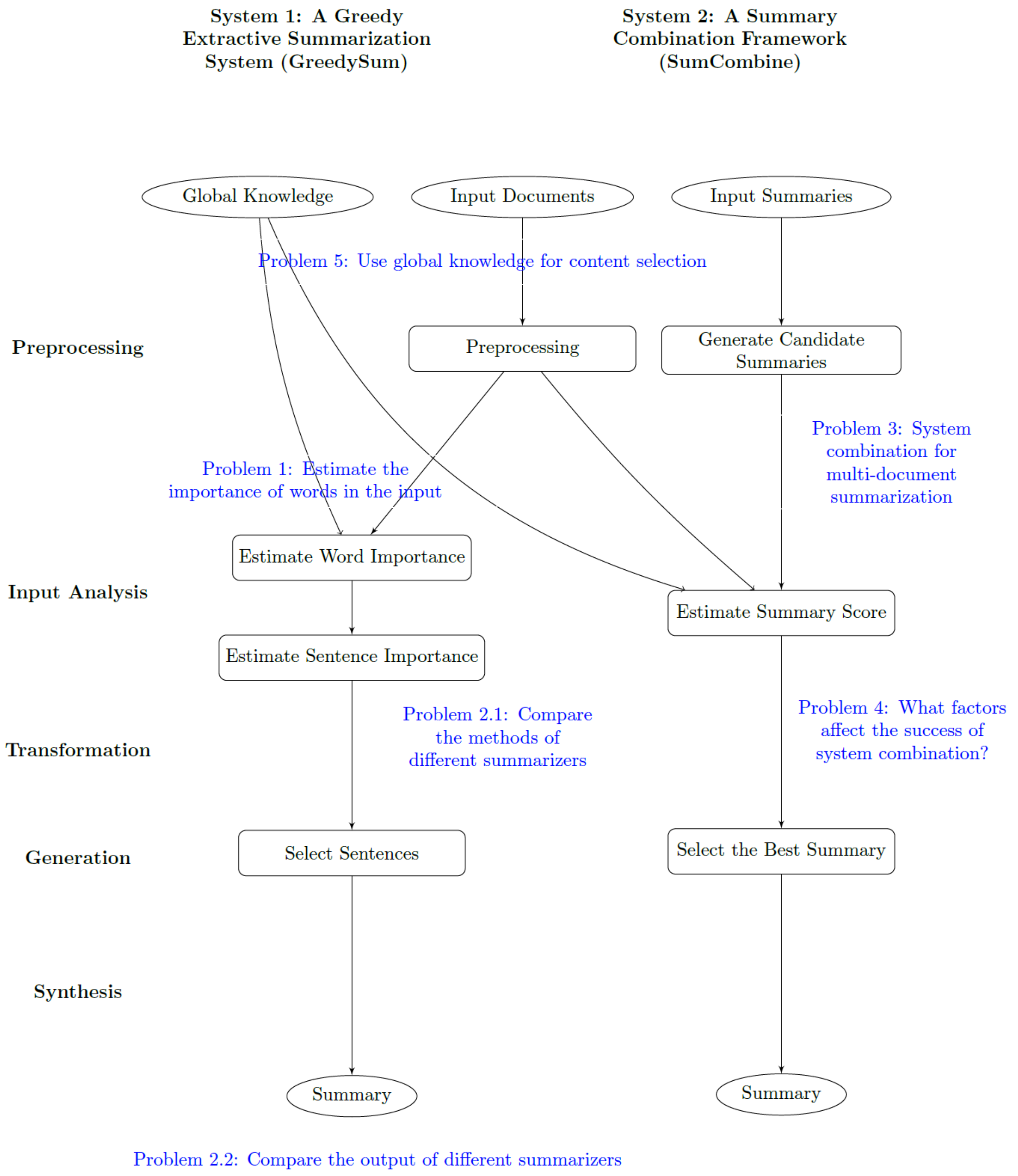

System 1: A Greedy Extractive Summarization System (GreedySum)
System 2: A Summary Combination Framework (SumCombine)
Global Knowledge
Input Documents
Input Summaries
Problem 5: Use global knowledge for content selection
Preprocessing
Preprocessing
Generate Candidate Summaries
Problem 3: System combination for multi-document summarization
Problem 1: Estimate the importance of words in the input
Estimate Word Importance
Input Analysis
Estimate Summary Score
Estimate Sentence Importance
Problem 2.1: Compare the methods of different summarizers
Problem 4: What factors affect the success of system combination?
Transformation
Generation
Select Sentences
Select the Best Summary
Synthesis
Summary
Summary
Problem 2.2: Compare the output of different summarizers

Figure 1.1: General framework of the two summarization systems proposed in this thesis. Documents or resources are shown in eclipses, processing stages are shown in rectangles. We also show the main problems that are tackled in this thesis.

al., 2013). Some compressive summarization systems directly select sub-sentences to be included in the final summaries, which combines the transformation and generation stages (Berg-Kirkpatrick et al., 2011).

In this thesis, we propose two summarization systems (see Figure 1.1). The first system (*GreedySum*) is a greedy extractive summarization framework (the left pipeline in Figure 1.1). During the input analysis stage, GreedySum first assigns the importance of words in the input, then estimates sentence importance based on word importance. During the generation stage, GreedySum iteratively selects sentences to be included in the summary based on their importance. The second system (*SumCombine*) conducts summary combination (the right pipeline in Figure 1.1). It first generates a pool of candidate summaries based on the summaries that are combined, then estimates informativeness of the candidate summaries. The best one is used as the final summary.

Based on the two systems, we investigate five areas that are concerned with improving the content quality of machine summaries (see Figure 1.1). First, we present a supervised model to improve the estimation of word importance in the input (*Problem 1*). We have the belief that doing so is helpful to generate better summaries. Second, we focus on comparing different summarization systems. The comparison focuses on two aspects: the summarization procedures (*Problem 2.1*) and the outputs (*Problem 2.2*). Clearly, knowing which system is better and why are critical for developing better systems. For *Problem 2.1*, we study how would different word weighting methods affect the final summary quality, where GreedySum is used as the summarizer. We show that by using the word weights estimated by *Problem 1*, GreedySum generates summaries with a performance comparable to the state-of-the-art. Third, we present a novel method that combines the summaries from different systems, which achieves considerable improvement over the systems that are combined (*Problem 3*). Fourth, we study what factors affect the success of system combination (*Problem 4*). The factors that we study include: diversity, performance, and combination methods. Fifth, we mine global knowledge with the help of dictionaries and external sources, which aims to facilitate the identification of informative content (*Problem 5*). The global knowledge based features are used for input analysis in both word importance estimation and summary combination.

Below we introduce the motivations, methodologies and findings of the five areas that we have described above. We introduce them in the order that they appear in the thesis chapters.

Improving the Estimation of Word Importance

Estimating word importance from the input document set (*input*) is critical for summarization systems. However, there has been little work that directly evaluates how well this is done for multi-document summarization. Indeed, even the systems that focus on word weighting were simply evaluated by their final summary quality. Different from prior work, here we focus on improving the estimation of word importance. Our work is based on the belief that improving word importance estimation is helpful to produce better summaries (we will show this in Section 4.1.3). Therefore, it is important to know how well the importance of words are estimated.

To evaluate the estimates of word importance, we introduce the task of identifying words from the input that are used in human summaries; we call such words *summary keywords*. Evaluation is performed by comparing the top words estimated by a word weighting method to the summary keywords.

The importance of words in the input are traditionally estimated using a single supervised method. However, our work is based on the hypothesis that better estimation of word importance can be achieved by combining various types of features. Evidence that supports this comes from our comparison of three unsupervised word weighting methods (word frequency, the log-likelihood ratio test, and a graph-based method), which have all proved to be effective in prior work. We find that the weights of words assigned by these methods do not have high correlations. This motivates us to explore and combine features derived by different methods.

We have explored a rich set of features. First, we include features related to frequency and location of the words. Second, we include part-of-speech and named entity tags of the words. These features have been widely used in keyword identification (i.e., identify keyphrases that are used for indexing) (Hulth, 2003), but are infrequently used in estimating word importance for summarization. Third, we include features that indicate whether a word appears in a high-quality machine summary. Fourth, we propose features that indicate the importance of the left-gram and right-gram of a word, based on the assumption that context around a word affects the importance of the word itself (Mihalcea and Tarau, 2004). Fifth, we employ two manually crafted dictionaries that group words into categories, based on the hypothesis that words in certain categories are likely or unlikely to appear in human summaries. Last but not least, we design features to indicate the intrinsic importance of words to humans. Especially, the last two classes of features estimate the importance of content based on information independent of the input, which are called *global knowledge*. We will discuss global knowledge in details in Chapter 5.

Combining the features above, we employ a logistic regression model to learn the word weights. The predicted keywords are the words with higher weights. Experiments show that our model improves the identification of summary keywords, compared to unsupervised methods and a supervised model that uses frequency and position related features. Though frequency of words in the input is the best method in identifying words that appear in all human summaries, the advantage of our model lies in its ability to better identify words that appear in one or two (out of four) human summaries; these words make up 85% of all summary keywords.

We investigate the effects of different feature classes by two experiments. First, we perform an ablation experiment that removes one class of feature at a time. Our experiment shows that frequency, location and word property features are very important. Ablating features related to intrinsic word importance or semantic categories leads to a small decrease in performance. Second, we train models that include traditional features (i.e., frequency and location) and one of the new feature classes. This experiment also shows that word property and intrinsic word importance features carry useful information.

Comparing Summarization Systems

The procedure of most summarization systems can be decomposed into two stages: a word weighting stage and a summary generation stage. Let I denote the input and let f denote the summarization process. The word weighting method and summary generation method are denoted as f_1, f_2 respectively. The process of producing the final summary S can then be represented as:

$$S = f(I) = f_2(f_1(I), I) \tag{1.1}$$

"*Most researchers build their own systems from the ground-up*" (Gillick, 2011), with different f_1, f_2, and even different preprocessing procedures. This makes it practically difficult to understand why one system is better than the other.

Here we fix f_2 and compare different f_1 towards the final summary quality. To find an optimal word weighting method (f_1), we present a modular greedy summarization system as our f (the left pipeline in Figure 1.1), which is simple, fast and transparent (Section 4.1.2). The "*word importance estimation component*" is regarded as f_1, the other components after that are regarded as f_2. Then we present a class of experiments that aim to answer the following questions:

- Which f_1 can lead to the best summary quality?
- Will better estimation of word importance lead to better summaries?
- Originally, words in the input are weighted by real values. However, it has been shown that for log-likelihood ratio weighting (Section 3.4.2) in particular,

weighing words using binary values (i.e., assign weight 1 to the keywords and weight 0 to the others) is better than using real values (Gupta et al., 2007). Can this idea be successfully extended to other word weighting methods?

To answer the first and the second question, we compare different word importance estimation methods. As it is difficult to evaluate different estimates of word importance, we use the evaluation result of summary keyword identification as a proxy. We show that the best method of identifying summary keywords is also the one that produces the best summary, which generates summaries comparable to the state-of-the-art on generic summarization of news. This answers the first question. However, we show that a model which can better identify summary keywords does not always lead to better summaries. Therefore, the answer to the second question is: most of the times, but not always.

To answer the third question, for each $f_1(w)$ whose output is a real value, we form a class of comparison groups $f_1'(w, k)$ by binary weighting: $f_1'(w, k)$ is equal to 1 if w is among the top k words ranked by $f_1(w)$, 0 otherwise. We show that binary weights work better if $f_1(w)$ is weighted by unsupervised methods, while real-valued weights work better if $f_1(w)$ is learned using our proposed model.

Our second study focuses on comparing the outputs from different summarization systems. This task seems easy, but is actually difficult because different systems were evaluated on different datasets, with different measures. To tackle these problems, we present a repository of summaries generated by six state-of-the-art and six competitive baseline systems on the Document Understanding Conference (DUC)[5] 2004 dataset, the most commonly used dataset for generic summarization. Moreover, our repository makes it feasible to compare these systems using paired test for statistical significance, which is recommended in Rankel et al. (2011). We show a surprising fact that the state-of-the-art systems generate very different summaries. This inspires us to study system combination in Chapter 6.

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

Extracting and Applying Global Knowledge

Summaries written by humans are inevitably influenced by their prior knowledge (Recht and Leslie, 1988; Endres-Niggemeyer and Neugebauer, 1998; Mani, 2001a). Inspired by this, we investigate the idea of using global knowledge—knowledge independent of a particular input—in summarization systems. We have the hypothesis that some content are globally important or unimportant to humans; we aim to identify such content.

In fact, this idea dates back to Edmundson (1969), where a list of pre-defined cue words are used to scores sentences in the input. Subsequent papers also utilize this idea for summarization in different domains (Pollock and Zamora, 1975; Paice, 1980; Paice and Jones, 1993; Schiffman et al., 2002; Woodsend and Lapata, 2012). However, the majority of summarization systems developed in the recent years ignore global knowledge and decide the important content solely based on characteristics of the input, such as frequency and locations of words. In this thesis, we explore two kinds of global knowledge: dictionary knowledge and intrinsic word importance.

First, we extract knowledge from dictionaries, based on the hypothesis that words in certain categories are likely to be included or avoided in human summaries. Here we use two dictionaries: a subjectivity and polarity lexicon (MPQA) and a dictionary that groups words into semantic and lexical categories (LIWC). The features extracted from these dictionaries are used for identifying *summary keywords*. Analysis of MPQA features shows that humans tend to avoid words of strong subjectivity. Analysis of LIWC features shows that humans tend to include words that belong to the semantic categories of “death”, “anger” and “money”.

Second, we assign intrinsic importance scores (*global indicators*) to individual words, derived by analyzing 160K summary-article pairs from the New York Times (NYT) corpus (Sandhaus, 2008). The words that are identified as important include “conflicts”, “hurricane”, while the words that are identified as unimportant include “slowly”, “Mr.”. To demonstrate the effectiveness of these global indicators, we build a summarizer (Blind) where the importance of words are simply decided based on these. Note that Blind does not conduct any analysis of the input. We show that Blind performs on par with a standard baseline in summarization, which selects the first L words from the most recent input document. In order to understand why some words are intrinsically important (unimportant), we study the categories that these words belong to in MPQA and LIWC (Section 5.4.2)—we think it is possible that these words are regarded as intrinsically important (unimportant) because they have certain semantic properties.

The two kinds of knowledge are used as features in our word importance estimation model, whose effectiveness is evaluated on summary keyword identification. Experiments show that ablating global knowledge leads to a small decrease in performance. We further examine the performance of GreedySum (the left pipeline in Figure 1.1), based on the estimated weights after global knowledge features are removed: we only observe a marginal decrease in performance. We will show in Chapter 6 that these global indicators are useful in identifying informative summaries for the summary combination task.

Finally, we show that the global indicators are effective in identifying low frequency summary keywords, where ablating them results in a significant decrease on this task (p-value < 0.001). However, removing global indicators does not affect the performance in identifying high frequency summary keywords. Indeed, for words that appear with low frequency in the input, abstractors may need to decide which ones should be used in summaries based on their prior knowledge.

System Combination for Multi-document Summarization

System combination has enjoyed great success in many natural language processing domains, such as automatic speech recognition (Fiscus, 1997), machine translation (MT) (Frederking and Nirenburg, 1994) and parsing (Henderson and Brill, 1999). In this thesis, we explore system combination for multi-document summarization.

This problem has been studied in a handful of papers. Based on the ranks of the input sentences assigned by different systems (a.k.a, *basic systems*), methods have been proposed to rerank these sentences (Wang and Li, 2012; Pei et al., 2012). However, these methods require the summarizers to assign importance scores to all sentences in the input. Thapar et al. (2006) combine the summaries from different systems, based on a graph-based measure that computes summary-input or summary-summary similarity. However, their method does not show any advantages over the basic systems. In summary, few prior papers have successfully generated better summaries by combining the outputs of different systems.

In our work, we focus on *practical system combination*, where we combine the summaries from four portable unsupervised summarizers. Our choice is based on three reasons. First, these systems are either off-the-shelf or easy-to-implement. This makes it easy to apply our method to new datasets. Second, even though many systems have been proposed for MDS, the outputs of them are available only on one dataset or even unavailable. Third, compared to more sophisticated supervised methods, unsupervised methods perform unexpectedly well. Many of them achieved the state-of-the-art performance when they were proposed and still serve as competitive baselines (see Section 4.2).

After the summarizers have been chosen, we present a two-step pipeline that combines summaries from the basic systems (*i.e., basic summaries*). In the first step, we generate candidate summaries. Here we study two methods to do this: (1) using the entire basic summaries directly, and (2) combining the basic summaries on the sentence level. We choose the second method, because the best possible performance of that method is much higher.

The second step selects among the candidate summaries. To do this, we propose a rich set of features that encode the content importance of the entire summary. Our features can be categorized into three classes: summary level, word level, and system identity features. Summary level features directly estimate the importance of the summary (e.g., input-summary similarity). Word level features estimate the overall importance of n-grams ($n = 1, 2$) in a summary. These features are equal to linear combinations of feature vectors of the n-grams in a summary, which encode the importance of these n-grams. System identity features capture the intuition that content from a better summarizer should have a higher chance to be selected. By ablation experiments, we show it is useful to utilize features derived based on different perspectives, from different sources (Section 6.7.3).

Based on these features, we employ a support vector regression model to score the summaries. Our model performs better than the best basic system when generating short summaries (i.e., 50, 100 words), which is comparable to the state-of-the-art systems on multiple benchmarks. However, our model does not outperform the best basic system when combining longer summaries (i.e., 200, 250, 400 words). We show that two major problems need to be tackled: (1) it becomes intractable to generate all candidate summaries, (2) it becomes practically more difficult to outperform the best basic system (Section 6.7.4).

Factors that Affect the Success of System Combination: An Empirical Study

We investigate how different factors affect the success of system combination. Researchers in machine translation shows that it is important for systems combined to have a better performance and high diversity (Macherey and Och, 2007; Cer et al., 2013; Gimpel et al., 2013). We are interested in whether similar claims can be made for multi-document summarization.

Our primary study focuses on *properties of the basic systems* (macro-level) (Section 7.2). We mainly focus on *performance* of the basic systems (the reason why we do not consider other factors are explained in the last paragraph of Section 7.2.1). This study is very useful, because oftentimes we need to decide whether system combination is necessary. We call the top performing basic system the *primary* *system*, the other basic systems *auxiliary systems*. Four combination methods are used here: (1) the model described in Chapter 6 (SumCombine), (2) summarization over summaries, (3) choose the summary with the highest overlap with all machine summaries, and (4) choose the summary with the smallest input-summary Jensen-Shannon (JS) divergence. We provide insights on what properties are important for designing good combination methods.

Our experiment reveals several findings. First, it is important to combine systems that have similar performance. When the primary system is fixed, performance of the combined system is positively correlated with mean performance of the auxiliary systems. If the basic systems perform similarly, all approaches outperform the primary system comfortably. Second, JS achieves a strong performance, which only falls behind that of SumCombine by a small margin. The other two methods perform poorly if the basic systems do not have similar performance; we discuss why this happens based on properties of the combination methods. Third, the improvement of combining low-performing systems is larger than combining top-performing systems. Hence, if a large improvement can be achieved by combining straw-man systems, we cannot safely claim that a good combination method has been developed. It is more convincing to show that a method can outperform the systems that are combined and advance the state-of-the-art.

Before our major study, we have also performed a preliminary study that focus on *properties of the basic summaries* (micro-level) (Section 7.1). There, we use the basic systems and the combination method in Chapter 6. We consider three factors (a.k.a, *explanatory variables*): overlap between the summaries, mean quality of the summaries, and difference in quality between these summaries. Because the input factors are correlated (Section 7.1.2), we compute zero-order and partial correlation between one of these input factors and *response variables* (the outcomes we want to study); partial correlation controls the other factors. For clarity, we only describe the result of partial correlation in the next paragraph.

We consider three response variables: quality of the oracle summary, potential improvement over the primary system, and real improvement over the primary system. Even though high *diversity* (low bigram overlap) is significantly correlated with a high oracle and a high potential improvement, it is not correlated with real improvement on our dataset. This is somewhat unexpected, given prior results in MT which suggests that diversity is critical. We observe that it is preferred to combine summaries of similar quality, which resembles our observation on the system level.

1.1 Thesis Organization

This thesis is organized as follows:

Chapter 2 describes the *news data* used in the thesis (Section 2.1). We then describe the Pyramid method and ROUGE, which are respectively the manual and automatic evaluation methods used in this thesis (Section 2.2).

Chapter 3 presents a superior supervised model for estimating word importance. We start with a review of prior work (Section 3.2) and compare three unsupervised methods of predicting word importance. We then describe the features used in our model (Section 3.5). Evaluation results are shown in Section 3.6.

Chapter 4 focuses on comparing different summarization systems. Section 4.1 compares different word weighting methods based on a modular greedy summarization framework. Section 4.2 introduces a repository of summaries we have collected from the state-of-the-art and competitive baseline systems. We evaluate and compare these summaries.

Chapter 5 describes the methods of mining and using global knowledge. We start with a literature review on global knowledge in Section 5.2. We then describe the knowledge extracted from dictionaries in Section 5.3 and introduce our method of estimating intrinsic word importance in Section 5.4. Experiments and results are shown in Section 5.5 and Section 5.6.

Chapter 6 presents our summary combination system. Section 6.4 discusses two strategies of generating candidate summaries: using the full summary outputs of the basic systems and reconstructing summaries by combining the outputs of basic systems on the sentence level. Section 6.5 presents the core component of our system: a supervised model that selects among the candidate summaries. Experiments and results are shown in Section 6.7.

Chapter 7 presents an empirical study on how different properties of the input basic summaries or summarizers affect the final summary quality. We focus on properties of the basic summaries in Section 7.1 and properties of the basic systems in Section 7.2.

Chapter 8 reviews the main findings in this thesis and discusses future directions.

1.2 Thesis Contribution

In this thesis, we make the following contributions:

Word importance estimation: We introduce a rich set of novel features that indicate the importance of words in the input documents. Apart from frequency and location based features, we include part-of-speech, named entity tags, reinforcement information from machine summaries, semantic categories, and intrinsic importance of the words; many of these features are not widely used in prior work on this task. To evaluate the estimates of word importance, we introduce the task of identifying words that appear in human summaries (*summary keywords*). Based on the proposed features, we present a logistic regression model to find summary keywords, which achieves better performance than prior methods of estimating word importance. We also introduce a novel method to evaluate summary keyword identification, inspired by the Pyramid method. We present a throughout study on the effects of features.

Comparing summarization systems: We study the effectiveness of different word weighting methods towards the final summary quality. We present a greedy extractive summarizer, whose modularity and transparency makes it easy to compare different weighting methods. There, we show that the best method of identifying summary keywords is also the best one for summarization, which achieves a performance comparable to the state-of-the-art for generic summarization of news. We also discuss whether real-value or binary weights should be used in summarization systems. We show that binary weights works better if the weights are estimated by unsupervised methods; while real-value weights works better if the weights are learned by our proposed supervised model.

We present a new repository that includes summaries from six state-of-the-art and six competitive baseline systems. Our repository addresses the problem that different systems were evaluated on different datasets, based on different ROUGE metrics. Our repository also makes it feasible to compare summaries from recent systems. Future researchers can also use summaries in our repository for comparison. Our experiments show that summaries from automatic systems have very low overlap in terms of content.

Extracting and applying global knowledge: We study the use of global knowledge, an aspect ignored in many summarization systems developed in recent years. We propose two methods of mining global knowledge: (1) using dictionaries that group words into categories, and (2) computing the intrinsic importance of words by analyzing the summary-article pairs from a large corpus. There, we find that certain words or categories tend to be globally important or unimportant. Moreover, we test the effects in three tasks related to content selection: summary keyword identification, summarization, and summary combination, where we observe a small improvement on all three tasks. In addition, we show that intrinsic importance of words are very helpful in identifying words that have low frequency in the input.

System combination: We present a new framework of system combination for multi-document summarization (SumCombine). We first generate candidate summaries by combining whole sentences from different systems. We show that summary combination is very promising, based on an analysis of the best choice among the candidates. To select among these candidates, we employ a supervised model that predicts the informativeness of the entire summary. Our model relies on a rich set of novel features that capture content importance from different perspectives, based on various sources. Ablation experiments verifies the efficacy of our features. SumCombine generates better summaries than the best summarizer while combining short summaries and achieves a performance comparable to the state-of-the-art on multiple DUC/TAC datasets. We also discuss why our model fails while combining longer summaries.

We investigate factors that affect the success of system combination. Our main study focuses on properties of the basic systems (macro-level). We show that it is critical to combine systems that have similar performance: if the basic systems perform similarly, even very simple methods outperform the basic systems; if some basic systems are much inferior than others, then methods based on consensus between summaries cannot achieve a good performance. We also show that for combination, it is easier to improve over low-performing basic systems than high-performing ones. This implies that a combination method that achieves a large improvement by combining low-performing systems might not be very effective. Moreover, we show that our model proposed in Chapter 6 is the most effective combination method, while selecting the summary with the smallest input-summary Jensen-Shannon divergence is a strong baseline.

We have also conducted a preliminary study based on the data and the basic systems in Chapter 6. This study focuses on properties of the basic summaries (micro-level). We observe a significant relation between diversity and mean quality of the basic summaries, which might be helpful to predict the difficulty of summarizing an input (Section 7.1.2). Surprisingly, we find that diversity is not a factor that affects the success of system combination on our data. Future research may investigate whether these findings hold in general.

Chapter 2

Data and Evaluation

We describe the data used in the thesis in Section 2.1 and the evaluation methods towards content selection quality in Section 2.2.

2.1 News Data from DUC and TAC

We focus on *multi-document summarization*, which produces a summary according to a set of related documents on a given topic. We mainly focus on *generic summarization*, where the task is to produce a summary according to the input documents. The topic is assumed to be not given during summarization. Another task that we also investigate is *topic-based summarization.* For the latter task, a system is given a set of documents as well as a topic statement. The system is expected to provide a summary that addresses the statement.[1]

We perform our analyses on data from the multi-document summarization task of the Document Understanding Conference (DUC) between 2001 and 2007 (Over et al., 2007) and from the Text Analysis Conference (TAC) in 2008 and 2009. These conferences are organized by the National Institute of Technology (NIST). The tasks

[1] Topic-based summarization can be regarded as a variation of *query-focused summarization* (Nenkova and McKeown, 2011), which generates a summary that answers a query. In this thesis, we use these two terms interchangeably.

Year	2001	2002	2003	2004
Number of input document sets	30	59	30	50
Number of documents per set	6–16	5–15	10	10
Number of human summaries	3	2	4	4
Summary length	50/100/200/400	50/100/200	100	100

Year	2005	2006	2007	2008	2009
Number of input document sets	50	50	45	48	44
Number of documents per set	25–50	25	25	10	10
Number of human summaries	4–9	4	4	4	4
Summary length	250	250	250	100	100

Table 2.1: Description of the generic (top) and topic-based (bottom) multi-document summarization datasets from the DUC 2001–2007 and TAC 2008–2009 workshops.

in DUC 2001–2004 are generic summarization of *newswire articles*, while the tasks in DUC 2005–2007 and TAC 2008, 2009 are topic-based summarization of *news*. We use the DUC 2003, 2004 dataset in Chapter 3, 4, 5 and all nine datasets in Chapter 6. The DUC 2001–2004 and TAC 2008, 2009 datasets are used in Chapter 7.

The summarization problem was created by experts who collected a group of related newswire articles on a same event. For topic-based summarization, the expert who collected the documents also create the topic statement. Then the automatic summarization systems generate summaries up to a certain number of words.[2] Summaries over the length limit will be automatically truncated.[3] To facilitate evaluation, NIST assessors create summaries that are about the same length as machine summaries. In Section 2.2, we will show how the quality of machine summaries are evaluated using the manually-generated summaries.

[2] An exception is the DUC 2004 evaluation, where summaries of up to 665 bytes (around 100 words) were required. This means systems will truncate words to different numbers of words for evaluation. This is disturbing, since the variation in length has an impact on automatic evaluation results. Therefore, later work (as well as our work) still mostly truncate the summaries to 100 word summaries during evaluation on this data.

[3] See http://http://www-nlpir.nist.gov/projects/duc/duc2005/ for a sample guideline

Topic: The 1998 NBA lockout

Human summary
In a dispute over a new collective bargaining agreement the National Basketball Association owners declared a lockout on July 1, 1998. They wanted to discard a clause in the old agreement allowing teams to pay their own free agents whatever they wanted, substituting a hard salary cap. The players wanted to keep earning as much as possible. On Oct. 5 all 114 preseason games were cancelled. The players then proposed a 50% tax on salaries above $18 million that the owners rejected. On Oct. 13 the NBA cancelled the first two weeks of the regular season. By Oct. 21 the entire season seemed in jeopardy in the interests of the best paid.

Machine summary
The decision to cancel 99 games between Nov. 3 and Nov. 16 came after the players association proposed the implementation of a tax system instead of a hard salary cap, a proposal the owners said they would respond to by Friday. In a critical ruling for the North American National Basketball Association and the players union, arbitrator John Feerick decides Monday whether more than 200 players with guaranteed contracts should be paid during the lockout. Last year, the players received about $1 billion dollars in salaries and benefits and we have made proposals that are guaranteed to increase that number by 20 percent over the next four years, Granik said in a prepared statement.

Topic: White supermacists
Narrative/Topic statement: Describe the widespread activities of the white supermacists and the efforts of those opposed to them to prevent violence.

Human summary
White supermacists often travel cross-country staging protests. An Arkansas-based group protested in Boston. A Virginia-based group marched in Toledo, Ohio. Neo-Nazis urged followers to travel to Crawford, Texas to protest the Iraq war. Oneneo-Nazi leader plotted to kill a federal judge. White supremacists spread their word through books and internet postings, often quoting King and other civil rights leaders to advance their own agendas. To avoid violent clashes, community leaders have pleaded for calm, staged peace rallies and delayed announcing protest routes. A national watchdog group monitors warehouse activities of online retailer Aryan Wear in Dallas-Fort Worth, Texas.

Machine summary
White supremacists clashed with an angry crowd outside Faneuil Hall, where Holocaust survivors and their families were commemorating the liberation of Nazi concentration camps. A white Republican lawmaker who contends he was excluded from a Black legislative group solely because of his race said, in September 2005, that the group is even more racist than the Ku Klux Klan. We were protesting black racial violence against white people in that neighborhood, said White. Navarre said the riots escalated because members of the National Socialist Movement took their protest to the neighborhood, which is predominantly black, instead of a neutral place.

Table 2.2: The human and machine summaries towards a generic and a topic-based summarization problem. Sample input documents of these two problems are provided in Appendix A.

Table 2.1 provides the basic statistics of our dataset: the number of input document sets (input), the number of documents per input, the number of human summaries per input and the length limit of the output. We also provide examples of human summaries and machine summaries towards a generic as well as a topic-focused summarization problem in Table 2.2. It is easy to tell from the example that the human summaries have better content and linguistic quality.

Note here that the DUC 2007, TAC 2008, 2009 shared tasks all include a main task and an *update summarization* task. The data we described are from the main task. The update task requires summarizers to produce summaries under the assumption that the abstractor has already read a set of earlier articles. Note also that we do not use the TAC 2010, 2011 data, because they are created for *guided summarization*: a task where the summarizer should produce summaries to include all aspects that are specified in the guidance of the summarization problem.

Our methods are evaluated on newswire articles. These methods may not be appropriate if the documents to be summarized are from other domains (e.g., medical records, legal text, meeting transcripts). Indeed, documents from different domains have different structures and properties. For example, for scientific articles, abstracts and conclusion often summarize the contribution of the paper. Systems that summarize a scientific article may also utilize the snippets that cite this article in other papers (Qazvinian and Radev, 2008; Mohammad et al., 2009; Xu et al., 2015). Systems that summarize articles in medical domains often utilize large-scale knowledge resources (e.g., Unified Medical Language System (UMLS) (Bodenreider, 2004)) to identify medical terms in the input documents. Such kind of domain specific information is helpful to identify information that should be included in the summary.

2.2 Evaluation

In this section, we mainly review two methods for evaluating the content in summarization: the Pyramid Method (Nenkova and Passonneau, 2004; Nenkova et al., 2007) and ROUGE[4] (Lin, 2004), which are used in this thesis. These two are the most widely used manual (the Pyramid Method) and automatic (ROUGE) evaluation methods, respectively. Both methods evaluate the quality of a summary by comparing the summary to several gold-standard human summaries (models).

There are also other methods to evaluate the content quality, such as the manual evaluation of *content responsiveness* and automatic evaluation without human models. We will discuss these methods in Section 2.2.3.

2.2.1 The Pyramid Method

The Pyramid Method solicits human annotators to score a summary based on its coverage of content that appear in multiple human models. Specifically, the Pyramid Method includes two steps: (1) making the pyramid from human models, and (2) scoring the summary (S). In the first step, annotators mark the content expressed in all human abstracts. Then they group the content with the same meaning into the same cluster, which is called a *Summary Content Unit* (SCU). Each SCU is assigned a weight according to the number of human models that this SCU appears in. In the second step, human annotators identify SCUs that are expressed in S. The *Pyramid score* of S is defined as the sum of weights towards SCUs that appeared in S divided by the weight of an ideally informative summary with k SCUs, where k is defined as the average number of SCUs across all human models.

Since first proposed, the Pyramid Method gained popularity in evaluating content. It was adopted by DUC and TAC as one of the official evaluation methods in 2005. The Pyramid Method has the following advantages. First, this method relies

[4] Recall-oriented understudy for gist evaluation

on SCU, a semantically driven sub-sentential unit. Clearly, compared to words or sentences, this is a more appropriate granularity of representing content. Indeed, a sentence often express multiple meanings, while words and phrases are too fine grained to capture what content is expressed. Second, this method considers semantic equivalence by grouping content with the same meaning from different human models into the same SCU. Third, it considers the human variations while writing summaries (Salton et al., 1997; Mani, 2001b) by using multiple models. Fourth, it has been demonstrated that this method makes the evaluation results less dependent on what human models are used as the gold standard (Nenkova et al., 2007).

The main concern of the Pyramid Method lies in the fact that it is very time-consuming. Therefore, researchers often choose to evaluate their systems using automatic methods, accompanied by an evaluation using the Pyramid Method on a small subset of the test data. We look into such kinds of automatic methods next.

2.2.2 ROUGE

The ROUGE (Lin, 2004) method is inspired by the success of BLEU[5] (Papineni et al., 2002) for machine translation. Both ROUGE and BLEU compute the n-gram overlap between the machine generated output and a set of human models. Different from BLEU that emphasizes precision, ROUGE is a recall-oriented metric. Because ROUGE is recall-oriented, it is important to truncate the summaries to a fixed length for evaluation, which can be automatically done by ROUGE.

A recall-oriented method is preferred for summarization due to two reasons. First, a summary can be regarded as a short article that covers the most important content, which are likely to be the ones that appear in human models. Therefore, evaluating the coverage of content that appear in human models is “recall” by definition. Second, the developers of ROUGE empirically showed that a recall-oriented metric

[5] Bilingual Evaluation Understudy

correlates better than a precision-oriented metric to the DUC coverage—the general accepted manual evaluation approach at that time (Lin and Hovy, 2003a) .

Formally, let $H_1, \ldots, H_k$ denote human summaries and let S denote the summary to be evaluated. The formula for ROUGE-n is:

$$\text{ROUGE-n} = \frac{\sum_{i=1}^{k} \sum_{t \in H_i} min(Count_{H_i}(t), Count_S(t))}{\sum_{i=1}^{k} \sum_{t \in H_i} Count_{H_i}(t)} \tag{2.1}$$

Here t is the n-gram that appears in H_i. $Count_{H_i}(t)$ and $Count_S(t)$ are the number of times t appears in H_i and S, respectively.

ROUGE has quickly become the de-facto standard evaluation tool of content quality because it is easy, fast and objective. There are many parameters that can be set for ROUGE, including decisions on whether or not to perform stemming, include stopwords, or truncate the summary, etc. It also includes many extensions beyond n-gram overlap, such as computing the overlap of longest common sequence (ROUGE-L) or basic elements (ROUGE-BE) (Hovy et al., 2006). However, the large number of parameters to choose has led to an undesirable situation: different researchers use different parameters while evaluation, which makes system comparison difficult. This is one of the problems that we aim to tackle in Section 4.2.

Even though automatic evaluation has been viewed as unreliable by some researchers, recent research shows that ROUGE-1 and ROUGE-2 can in fact emulate the preference of human annotators of one system over the other with at least 87% accuracy (Owczarzak et al., 2012). In this thesis, we employ three ROUGE metrics: ROUGE-1, -2, -4, all with stemming and stopwords included.[6] The reasons why we use ROUGE-1, -2, -4 is as follows. First, suppose there are two systems A and B, which are assigned scores (e.g. Pyramid score) by human assessors. Owczarzak et al. (2012) show that ROUGE-2 can predict whether A/B is (significantly) better than B/A with the highest accuracy, among all ROUGE metrics they examined. Second,

[6] ROUGE-1.5.5 with the parameters: -n 4 -m -l 100 -x -a. -n 4 specifies to compute 1-4 grams, -m specifies to perform stemming, -l 100 specifies to truncate to 100 words, -x means not to compute ROUGE-L, -a specifies to evaluate all reference systems.

suppose A is significantly better than B in terms of Pyramid score. Rankel et al. (2013) show that ROUGE-1 is the automatic evaluation metric that can predict this with the highest recall (i.e., the most sensitive metric), while ROUGE-4 can predict this with the highest precision (i.e., the most reliable metric).

2.2.3 Other Evaluation Methods

Apart from the Pyramid Method, *content responsiveness* is another popular method for evaluating content quality, which was employed as the official evaluation metric in DUC 2005–2007. Content responsiveness[7] evaluates the amount of information that helps to satisfy the information required in the topic statement. To get the score, annotators are first given a set of summaries, including both machine and human summaries. They will read these summaries, then grade each of them on a scale of 1 to 5 according to its relative informativeness compared to the other summaries.

Doing manual evaluation using content responsiveness is fast. However, it has several limitations. First, it is too coarse grained. In contrast to the Pyramid Method which employs SCUs to score a summary, the summary is directly graded. Second, it is subjective, as different annotators have different definition of "what is informative". Third, the responsiveness of the current summary depends on what summaries are given to annotators for comparison. Fourth, this method is specifically designed for query-focused summarization. Because of these, we still use the Pyramid Method for manual evaluation.

Recently, automatic evaluation methods that do not require human summaries have been developed (Louis and Nenkova, 2009; Saggion et al., 2010; Louis and Nenkova, 2013). Specifically, a method that compares the summary with the consensus of output from automatic summarization systems generates fairly accurate rankings, achieving a correlation of above 0.9 with the Pyramid score. However, this

[7] There are two kinds of responsiveness, content and overall responsiveness. The overall responsiveness considers both linguistic quality and the content.

evaluation method is not widely used in research papers. We will discuss more on this in Section 6.2, where we describe how this task is related to summary combination.

Chapter 3

Improving the Estimation of Word Importance

3.1 Introduction

Estimating[1] word importance from the input document set (i.e., *input*) is crucial for summarization systems. Extractive summarization systems (i.e., systems that identify the most important *sentences* in the input) estimate sentence importance based on the importance of words that appear in the sentence, along with other features such as sentence length and sentence position (Nenkova and Vanderwende, 2005; Conroy et al., 2006a; Ouyang et al., 2011). Compressive summarization systems (e.g., systems that compress input sentences) also often involve analyses of word importance, which are helpful in deciding which words or phrases in the input should be included in the final summary (Berg-Kirkpatrick et al., 2011; Almeida and Martins, 2013; Li et al., 2013a).

Even though many systems include a component that estimates word importance, there has been little work that empirically evaluates how well this is done. Indeed, even the summarization systems that focused on word weighting were simply

[1] This chapter is adapted from Hong and Nenkova (2014a) and Hong and Nenkova (2014b).

evaluated by their final summary quality. In contrast to prior work, we empirically evaluate the estimation of word importance. Our motivation is based on the hypothesis that improving the estimation of word importance is helpful in producing better summaries. Therefore, it is critical to know how well the words are estimated.

To examine the estimation of word importance, we introduce the task of identifying words from the input that are included in human summaries; we call such words *summary keywords*. Evaluation is thus performed by comparing the top words estimated by a word weighting method to the *summary keywords*. Another reason why we evaluate the accuracy of summary keyword identification is as follows. Rather than assigning weights to all words during summarization, some systems (Conroy et al., 2004; Conroy et al., 2006a) first identify a set of keywords and use those keywords to produce summaries. For this kind of system, improving keyword identification is directly related to improving summary quality.

The main focus of this chapter is to study how one can better estimate the importance of words in the input. Traditionally in summarization, this is done by using a single unsupervised method, such as word frequency (Luhn, 1958), TF*IDF (term frequency multiplied by inverse document frequency) (Sparck Jones, 1972) or the log-likelihood ratio (LLR) test (Dunning, 1993). However, our work is based on the hypothesis that one can achieve better estimation of word importance by combining various types of features. Later in this chapter (Section 3.4.4), we will give an evidence that supports this hypothesis—we will show that the weights of words assigned by three classical word weighting methods (word frequency, the LLR test, and a graph-based method) do not have high correlations. This inspires us to explore and combine features derived by different methods to predict word importance.

Motivated by the observation above, we explore a rich set of novel features. We also analyze the predictive power of these features using statistical tests. Specifically, we include the following types of features. First, we include features derived from unsupervised word weighting methods and word locations. Second, we include features concerning word properties, including part-of-speech tags, named entity tags and capitalization information. These features are not widely used in estimating word importance for summarization. Third, features are set to indicate whether or not one word appears in a machine summary, which helps to capture reinforcement information from these summaries. Fourth, we propose features that indicate the importance of the left-gram and right-gram of a word. This is based on the assumption that context importance around a word affects the importance of the word itself (Mihalcea and Tarau, 2004). Fifth, we include unigram features. Finally, we extract knowledge from resources independent of the input, which we call *global knowledge*. Here, we include subjectivity, topic categories mined from dictionaries and intrinsic word importance estimated from a large corpus.

Combining the features above, we present a logistic regression model to learn the final word weights. Experiments show that our model achieves better performance in identifying summary keywords, compared to other methods of estimating word importance. Specifically, the advantage of our model lies in its ability to better identify words used by one or two (out of four) abstractors. These words make up 85% of all summary keywords. However, we find that word frequency (probability) in the input set is the best method in identifying words that appear in all human summaries.

We then study the effects of different features. First, we conduct an ablation experiment, which shows that the frequency, location and word property features are the most important. Removing the features related to intrinsic word importance and semantic categories only leads to a small decrease in performance. Second, we train models that only include frequency, location and one of the new feature classes. Experiments show that the word property features are very useful, the intrinsic word importance features are somewhat useful, while the others are not that useful. The context features negative affects the performance, according to both experiments.

This chapter is organized as follows. In Section 3.2, we review prior work related to word weighting in summarization. In Section 3.3, we describe the summary keyword identification task. Section 3.4 describes three unsupervised methods of assigning importance to words and compares their differences. Section 3.5 describes the rich set of features used in our logistic regression model. Experiments and results are presented in Section 3.6, followed by a conclusion.

3.2 Related Work

The idea of identifying keywords that are descriptive of the input documents was first proposed in Luhn's fundamental work in automatic summarization (Luhn, 1958). There, keywords were identified based on the frequency in the input, where words that appeared most and least often were excluded. The sentences in which the keywords appeared close to each other were then selected to form the final summary.

Many successful recent systems also employ unsupervised methods to estimate word importance. For example, estimating word importance based on frequency (probability) has been rejuvenated in the past ten years, and summarizers based on this achieves impressive performance (Nenkova and Vanderwende, 2005; Nenkova et al., 2006). Apart from word frequency, words are also estimated using document frequency (DF) (Schilder and Kondadadi, 2008). Systems that optimize the coverage of bigrams also often use DF to weigh bigrams (Gillick et al., 2008; Gillick and Favre, 2009). Word importance is also often weighted by TF*IDF. Many extractive summarization systems represent the sentences into vectors using bag-of-word models, where each word is weighted by TF*IDF (Erkan and Radev, 2004; Radev et al., 2004b; McDonald, 2007; Lin and Bilmes, 2011). These systems then estimate the importance of sentences based on the vectors.

Some other methods assign weights to certain words, instead of all words in the input. A powerful method is the log-likelihood ratio test (Lin and Hovy, 2000), which identifies a set of words that appear more often in the input than in a background corpus; the authors call such words *topic signatures*. Lin and Hovy (2000) show the effectiveness of this method over TF*IDF for word weighting in single-document summarization. Later systems employ topic signatures to score sentences for multi-document summarization and achieve competitive performance (Conroy et al., 2004; Harabagiu and Lacatusu, 2005; Conroy et al., 2006a). Han et al. (2015) use a quantitative statistical method to discover salient keywords which characterize the contributions of a paper from a paper's citation history. The authors then use the identified keywords to develop a scientific paper summarization system. Graph-based approaches have also been used for extracting keywords for single-document summarization. There, the graph is constructed based on the co-occurrence of words in a sentence, and the HITS algorithm (Kleinberg, 1999) is used to rank these words (Litvak and Last, 2008).

Word importance has also been estimated by supervised approaches, with word frequency and position of word occurrence in the input (e.g., first occurrence of a word) as the most typical features. For example, a summarizer that weighs words using position and frequency outperforms using frequency alone (Yih et al., 2007); a stack decoding algorithm with performance guarantee can be used to maximize the coverage of word weights (Takamura and Okumura, 2009), where the estimation of word weights resembles that of Yih et al. (2007). Recently, a submodular function based framework has been developed to optimize the coverage of word weights (Sipos et al., 2012), where capitalization, word length, unigrams along with frequency and location are used as features. In general, however, the features that have been explored for estimating word importance in summarization are limited.

A handful of papers have productively explored the mutually reinforcing relationship between word and sentence importance. There, graph-based approaches are widely used to iteratively improve the estimation of word and sentence importance, in unsupervised (Zha, 2002; Wan et al., 2007; Wei et al., 2008) and later supervised frameworks (Liu et al., 2011).

Instead of estimating word importance, most prior work has aimed at scoring sentences in the input. Popular unsupervised approaches estimate sentence importance based on its similarity with the input (Centroid) (Radev et al., 2004b), average word frequency (Nenkova et al., 2006), latent semantic analysis (Gong and Liu, 2001; Steinberger and Jezek, 2004), and graph-based methods (Erkan and Radev, 2004; Mihalcea, 2005). Researchers have also extensively explored features used in supervised systems for estimating sentence importance. Earlier work (Kupiec et al., 1995; Teufel and Moens, 1997) used features such as sentence length and sentence position in a paragraph, later work combined a variety of indicators of sentence importance (e.g., sentence importance scores estimated by unsupervised methods) in their models (Litvak et al., 2010; Ouyang et al., 2011; Wang et al., 2013).

Prior work has also extensively investigated keyword (or rather keyphrase) identification[2] (e.g., extract a list of phrases from a journal article) , without further integrating the selected keywords in a full-fledged summarizer. Turney (1997) showed that word frequency in the input is a poor selector for keyphrases. A handful of work (Frank et al., 1999; Turney, 2000; Hulth, 2003) explored using supervised methods for this task, where the features used in those work include TF*IDF and position of the first appearance of the words (Frank et al., 1999), word frequency, document frequency, number of words in the phrase (Turney, 2000) and part-of-speech (Hulth, 2003). Competitive to these are unsupervised methods in which the weights of words are derived by a graph describing the co-occurrence of words in the same sentence in a fixed width window of context (Mihalcea and Tarau, 2004; Liu et al., 2010). One of our baseline methods is similar to the method of Mihalcea and Tarau (2004), but we estimate word weights for summarization.

[2] The task there is in fact keyphrase extraction, because the “keywords” are often phrases of at least two words.

Overall, only a handful of studies investigate suitable features for estimating the importance of words for summarization. In our work, we incorporate a much broader set of features for estimating word importance. Some of our features are derived from unsupervised methods, some are inspired by features used in keyphrase identification, some others are derived based on knowledge independent of the input.

3.3 Our Task: Summary Keyword Identification

To examine our prediction of word importance, we present the task of identifying *summary keywords*, defined as the words in the input that appear in human summaries. Here we use the DUC 2003, 2004 datasets, where input set includes 10 documents on the same topic. Four human summaries are provided for each input (Table 2.1). Since words that appear in different numbers of human summaries should be of different importance, we define different gold-standards G_i of summary keywords, where G_i represents the set of words that appear in at least i $(1 \leq i \leq 4)$ human summaries. Here we have $G_4 \subseteq G_3 \subseteq G_2 \subseteq G_1$: the words in G_4 are the most important ones, G_3 includes some less important words that do not appear in G_4, etc. Note here that G_i only include content words[3] that also appear in the original input. Among all content words that appear in human summaries, 27.3% (16.7% with stemming) of them never appear in the input.[4] Table 3.1 shows the average number of summary keywords in G_1, G_2, G_3 and G_4 in our data.

Once the importance of words has been estimated, we derive the predicted keyword list L by taking the top k words with the highest importance. Two approaches

[3] The stopword list is derived from the SMART system (Salton, 1971), augmented with punctuation and symbols.

[4] In the original work (Hong and Nenkova, 2014a), we use a tool designed to generate topic words (`http://homepages.inf.ed.ac.uk/alouis/topicS.html`) to generate our content words. It tokenized texts by stripping away all non-letters, then use spaces to split words. That method was relatively coarse, and would bring noise. In the thesis, we tokenized our text using Stanford CoreNLP (Manning et al., 2014), then generate the content words. Also we incorrectly included some contractions in the content word lists in the training data of the original work. Here we exclude those contractions. Hence, the results are slightly different from the original work.

Average $\lvert G_i \rvert$	1	2	3	4
DUC 2003	107.8	37.1	15.8	6.5
DUC 2004	108.8	34.3	15.5	6.3

Table 3.1: Average number of words in G_i on the DUC 2003, 2004 data

are used for evaluation. First, we compare the keyword list L to different G_i using F_1-measure[5]. However, while comparing L to G_1 (or G_2, G_3), F_1-measure does not consider the fact that words in different number of human summaries should be of different importance. To tackle this problem, we propose the *Keyword Pyramid Method* (see Section 3.6.2). Moreover, this method generates a single score, which makes the evaluation simpler.

3.4 Unsupervised Word Weighting

Here we describe three unsupervised methods of weighting word importance. The first two are probability and the log-likelihood ratio test, which have been extensively used in prior work. We also employ a Markov Random Walk model for word ranking, extended from Mihalcea and Tarau (2004). We finally discuss the correlations and differences between the words ranked by these three approaches.

3.4.1 Word Probability (Prob)

Word probability (frequency) is often used for estimating word importance. Let $c(w)$ denote the number of times word w appears in the input and let N denote the number of word tokens in the input, the word probability of w is $p(w) = c(w)/N$.

[5] 2*precision*recall/(precision+recall)

3.4.2 The Log-likelihood Ratio Test (LLR)

The log-likelihood ratio (LLR) test approach (Lin and Hovy, 2000) identifies topic words by comparing the distribution of words in the input (I) to that in a large background corpus (B).[6] We compute the LLR between two hypotheses:

Hypothesis 1 (H_1) : $P(w|I) = p = P(w|B)$, i.e., w is not a topic word.

Hypothesis 2 (H_2) : $P(w|I) = p_1 \neq p_2 = P(w|B)$, i.e., assuming $p_1 > p_2$, which indicates w is a topic word.

Here $P(w|I)$ and $P(w|B)$ are the probability that w appears in I and B, respectively. H_1 indicates that w is not a topic word, which appears in I and B with equal frequency. H_2 indicates that w is a topic word, which appears with higher frequency in I than in B. In our experiments, the stopwords are excluded from I and B before computation: only content words are considered.

Now we describe how the likelihood of the two hypotheses are computed. Consider a word w which appears in a set of documents with x probability. Let N denote the number of word tokens in these documents. The likelihood that w appears k times in N trails is a binomial distribution:

$$b(k; n, x) = \binom{n}{k} \cdot x^k \cdot (1 - x)^{n-k} \tag{3.1}$$

Let c_I, c_B denote the number of times w appears in I and B, respectively. Let N_I, N_B denote the total number of word tokens in I and B, respectively. The likelihood of H_1 and H_2 are:

$$L(H_1) = b(c_I; N_I, p) \cdot b(c_B; N_B, p) \tag{3.2}$$

$$L(H_2) = b(c_I; N_I, p_1) \cdot b(c_B; N_B, p_2) \tag{3.3}$$

[6] We use the topic word tool via this link: http://homepages.inf.ed.ac.uk/alouis/topicS.html. We use 10K documents in the New York Times corpus (Sandhaus, 2008) as the background corpus, which is different from what was originally provided in the toolkit.

Here the probability estimates are: $p = (c_I + c_B)/(N_I + N_B)$, $p_1 = c_I/N_I$, $p_2 = c_B/N_B$. The log-likelihood ratio between the two hypotheses is:

$$v = -2 \cdot \log \frac{L(H_1)}{L(H_2)} \tag{3.4}$$

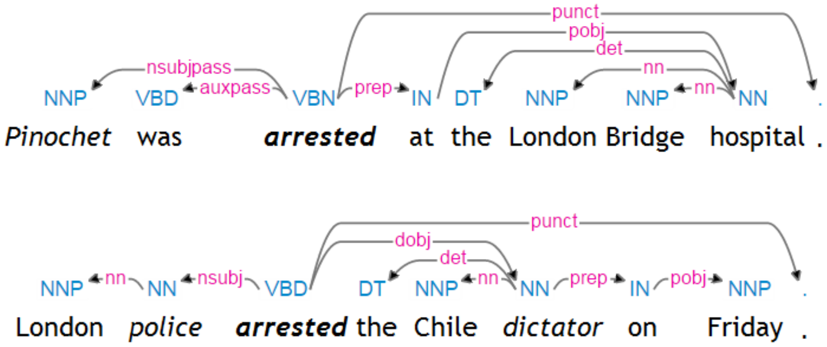
The LLR test provides a way of weighting words in the input, where words with higher importance tend to a have larger v. Since v is asymptotically χ^2 distributed (Lin and Hovy, 2000), it is natural to use the LLR to select keywords: one can first pick a confidence level (e.g., 95%), then get the value of the cutoff corresponded to the confidence level by looking up in a χ^2 distribution table. Words with weights above the cutoff are regarded as keywords.

Note that Lin and Hovy (2000) made an approximation while computing of $L(H_2)$: they assume $p_1 \neq p_2$ indicates $p_1 > p_2$. However, the authors (and many paper afterwards) did not explain why this assumption can be made. Noticing this, we give an explanation below.

While identifying topic words, the stopwords are excluded. This means that for a content word w, it is unlikely to appear in the background corpus with a very high probability (p_2). Even for the rare cases that w has a large p_2, it is unlikely for p_1 to have a small value ("nation", "united" are examples of content words with a large p_2). Therefore, for the words with $p_1 < p_2$, $L(H_1)$ and $L(H_2)$ are not likely to differ by a great extent, which means the LLR is unlikely to be too large. This hypothesis is verified by our experiments: if we use the cutoff $c = 10.0$ (corresponding to a 99.84% confidence interval), all topic words identified on the DUC 03, 04 data have $p_1 > p_2$. Here we still use the method of Lin and Hovy (2000) to compute the LLR ratio. For future work, we suggest to regard $p_1 < p_2$ and $p_1 > p_2$ differently, e.g., use v when $p_1 > p_2$ and use $-v$ when $p_1 < p_2$, which might give a more accurate estimation of word weights.

3.4.3 Markov Random Walk (MRW)

Graph-based methods (e.g., PageRank) are often used to weigh sentences in both generic (Erkan and Radev, 2004; Wan and Yang, 2008) and query-focused summarization (Otterbacher et al., 2009). In our work, we employ graph-based methods to weigh words instead of sentences. Our method is based on the assumption that a word syntactically related to other important words is likely to be important. For instance, consider the following two sentences shown in Figure 3.1, where the words *Pinochet*, *dictator* and *police* are all important for the input. Since they are syntactically related to the verb *arrested*, we assume that this verb is also likely to be important (i.e., included in human summaries).[7]



punct
pobj
det
nsubjpass
nn
auxpass
prep
nn
NNP VBD VBN IN DT NNP NNP NN .
Pinochet was ***arrested*** at the London Bridge hospital .
punct
dobj
det
nn
nsubj
nn
prep
pobj
NNP NN VBD DT NNP NN IN NNP .
London *police* ***arrested*** the Chile *dictator* on Friday .

Figure 3.1: Dependency relation of two sentences.

Graph-based methods provide a way to utilize this idea. Our method is similar to that of Mihalcea and Tarau (2004), which employed graph-based methods to identify keywords. However, there is one major difference between our method and theirs. Consider a graph where each node corresponds to a word type in the input. In Mihalcea and Tarau (2004), there are edges between two nodes iff these words co-occur within a window of k in a sentence. In our work, there are edges between two nodes iff these words are connected by a directed edge in the dependency tree.[8] An advantage of our method is that we do not need to tune k. Moreover, we build

[7] This graph is generated via this link: `http://demo.ark.cs.cmu.edu/parse`

[8] Here we use the Stanford Dependency Parser (De Marneffe et al., 2006).

edges based on dependency relations rather than word co-occurrence, which is a more fine-grained way to represent word relations.

Formally, let G denote the directed graph and let $v(w)$ denote the node corresponded to the word w. The initial unnormalized edge weights $d'((v(w_1), v(w_2)))$ are equal to the total number of syntactic dependencies between two words w_1, w_2 within the same sentence in the input. The final edge weight $d((v(w_1), v(w_2)))$ is equal to $\frac{d'((v(w_1),v(w_2)))}{\sum_{v(w) \in G} d'((v(w_1),v(w_2)))}$ in G. Here we do not necessarily have $d((v(w_1), v(w_2))) = d((v(w_2), v(w_1)))$ in the final directed graph G after the normalization step.

After G has been constructed, we apply the Markov Random Walk (MRW) algorithm to estimate word weights (Wan and Yang, 2008). This algorithm is similar to the PageRank algorithm (Page et al., 1998). However, it does not specify the outgoing edges of one node to have the same weight, which is different from the original PageRank algorithm. The main idea is that the importance of each node (word) can be estimated based on the votes from its adjacent nodes (words) in the graph. Formally, let $p(v)$ denote the importance of node v. This algorithm iteratively updates $p(v)$ until all $p(v)$ converge to a stable value. Here the edge weight $d((v(w_1), v(w_2))$ can be regarded as the transition probability from w_1 to w_2, where the process is a Markov chain. The final $p(v)$ is the stationary probability distribution of the Markov chain. The pseudo code of this algorithm is as follows:

Algorithm 1 The Markov Random Walk Algorithm

```
procedure MRW($G$, $d$)
    $p(v) = \frac{1}{|V|}, \quad \forall v \in V$
    while (not (for all $v$, $|p(v) - p'(v)| < \epsilon$)) do
        $p'(v) \leftarrow p(v), \quad \forall v \in V$
        $p(v) = (1 - \lambda) \cdot \sum_{v' \in V, v' \neq v} (p'(v') \cdot d((v', v))) + \lambda \cdot \frac{1}{|V|}, \quad \forall v \in V$
    end while
    return $p(v)$
end procedure
```

At each iteration, there is probability of λ to perform random jumps between nodes. Here we set $\lambda = 0.15$, following Wan and Yang (2008). The algorithm terminates when the change of node weights between iterations is smaller than ϵ for all nodes. We set $\epsilon = 10^{-4}$, following Mihalcea and Tarau (2004). The importance of w is equal to the weight $p(v(w))$.

3.4.4 Comparison of the Unsupervised Approaches

We are interested in whether the three unsupervised methods (Prob, LLR, MRW) rank words in a similar way. Indeed, suppose the ranked word lists have high correlation with each other, it might not be useful to combine the weights estimated by these methods.

Here we compute the Spearman correlation between the word weights assigned by these three methods for each input of the DUC 2003, 2004 data. Figure 3.2 demonstrates the boxplot of the Spearman correlation on these two datasets. Among the three comparisons, the ranked word lists produced by Prob and MRW are the most similar (median = 0.656, 0.646), followed by Prob and LLR (median = 0.465, 0.459), while LLR and MRW are the least similar (median = 0.273, 0.282). Clearly, the ranks assigned to words differ considerably from each other.

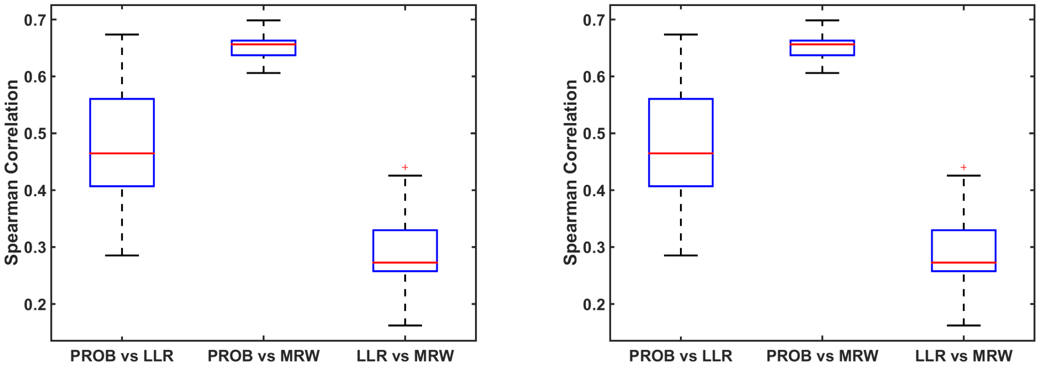


Figure 3.2: Spearman correlation between the word weights assigned by unsupervised approaches on the DUC 2003 (left) and DUC 2004 (right) data.

Here we provide examples of the top 20 keywords identified by these three methods for two inputs: topic d30020t (denoted as input A) and d31050t (denoted as

Input A

Prob	Rank 1-7:	games, China, Asian, Chinese, team, world, gold
	Rank 8-14:	women, athletes, competition, Japan, won, Xiong, Thailand
	Rank 15-20	championships, silver, Korea, newspaper, Chen, sports
LLR	Rank 1-7:	games, Asian, China, Chinese, xiong, gold, athletes
	Rank 8-14:	medalist, Thailand, medals, Xinhua, Hiroshima, Bangkok, team
	Rank 15-20	Chen, golds, competition, championships, medley, bronze
MRW	Rank 1-7:	games, team, athletes, won, China, championships, competition
	Rank 8-14:	gold, Chinese, make, medals, Korea, medalist, events,
	Rank 15-20	win, women, medal, committee, scored, Asian

Input B

Prob	Rank 1-7:	party, China, Wang, Qin, Xu, democracy, rights
	Rank 8-14:	political, dissidents, Chinese, years, government, trial, human
	Rank 15-20:	Liu, state, detained, Prison, democratic, police
LLR	Rank 1-7:	Qin, Xu, Wang, China, party, dissidents, democracy
	Rank 8-14:	Liu, dissident, detained, Youcai, Wenli, rights, subversion
	Rank 15-20:	Yongmin, Chinese, communist, Hangzhou, trial, Wuhan
MRW	Rank 1-7:	party, detained, China, dissidents, rights, Wang, Qin
	Rank 8-14:	government, Xu, years, trial, released, arrested, members
	Rank 15-20:	leader, wife, activities, dissident, signed, movement

Table 3.2: Top 20 words ranked by three unsupervised approaches on two input sets. Sample input documents of these two inputs are provided in Appendix A.

input B) of the DUC 2003 data (Table 3.2). This would give us an intuition of what kind of words are likely to be ranked high by each method.

Words that appear more often in the background corpus have lower ranks using LLR, compared to Prob. For example, the ranks of the words *world* and *team* decrease for the input A; the ranks of the words *party*, *government* and *political* decrease for the input B. Meanwhile, rare words get ranked higher by LLR. For instance, the rank of *medalist* gets promoted greatly for the input A, from out of 20 to 8; the ranks of the Chinese names *Qin*, *Xu* for the input B get promoted as well.

Words that are syntactically related to other important words are rewarded using MRW, compared to Prob and LLR. Examples include verbs such as *won, make, win* from the input A and *detained, released* from the input B; as well as nouns such as *athletes, championships* from the input A and *members, activities* from the input B.

In summary, the ranked word list generated by these three methods do not have as high correlation as expected. This inspires us to combine features derived from these methods, along with features that carry indicators from other perspectives to predict word importance for summarization.

3.5 Features

This section presents the rich variety of features included in our logistic regression model. We have at our disposal abundant data for learning because each content word in the input can be treated as a labeled instance. There are in total $33,023$ samples from the 30 inputs of DUC 2003 for training, $57,049$ samples from the 50 inputs of DUC 2004 for testing. For each word in the input, we assign label 1 if the word appears in one of the human summaries, otherwise we assign label 0. Note however the predicted word importance is real-valued.

Apart from describing the features, we also analyze the predictive power of these features on the DUC 2003 dataset. Here we conduct two tests: *Wilcoxon rank-sum test* (unpaired, a.k.a., Mann-Whitney U test) (Mann and Whitney, 1947) and *proportion test*. For a specific feature, we conduct tests between the feature vectors of the words that are used and not used in human summaries. Wilcoxon rank-sum (WRS) test in conducted for all features (i.e., binary and numeric features). The null hypothesis is that the two groups of words have the same mean value. We use WRS test (a non-parametric test) rather than a parametric test, because we are unsure whether or not our data is normally distributed. Proportion test is only conducted for binary features. The null hypothesis is that the two groups of words have the same proportion of ones for a specific feature tag. We regard proportion test as the primary statistical test for binary features.

There are in total $8,671$ features, of which $1,077$ are significant (p-value < 0.05), 557 are highly significant (p-value < 0.001) (Wilcoxon rank-sum test). We rank our features by increasing p-values by WRS test. Interestingly, some less explored features are significant or even highly significant.

3.5.1 Classical Word Importance Estimation Methods

This class of features is related to the frequency and location of a word, and have been widely used for estimating word importance in prior work.

Probability, LLR and MRW: For each word, we use its probability, LLR weights and MRW weights as features. The features are normalized according to their maximum value for an input. We also design features to indicate whether or not one word is ranked within the top k according to these three weighting methods. This is inspired by prior work which demonstrated that for LLR weights in particular, it is useful to identify a small set of important words (keywords) and ignore all other words for summarization (Gupta et al., 2007). Specifically, the value of the feature is 1 if one word is ranked within top k, 0 otherwise. Here k are preset cutoffs[9] that represent different ways of defining important words in the input. There are 3 real-valued features and 96 binary features in this class, all of which are highly significant, ranked within the top 180 most significant ones by WRS test ($p < 10^{-113}$).

Document Frequency: For each word, we include a feature that equals to the document frequency of the word in the documents to be summarized. This feature is very significant, ranked within the top 100 by WRS test ($p < 10^{-300}$).

Word Locations: Especially in newswire articles, sentences that appear at the beginning are often the most important ones (Edmundson, 1969). In line with this observation, we compute several features related to the position in which a word appears. We first compute the relative positions for word tokens in one document, where the tokens are numbered sequentially in order of appearance in each document

[9] 10, 15, 20, 30, $\cdots$, 190, 200, 220, 240, 260, 280, 300, 350, 400, 500, 600, 700 (in total 32 values)

in the input. The relative position for one word token is thus equal to its corresponding number divided by the total number of tokens minus one in the document, i.e., 0 for the first token, 1 for the last token. For each word type, we calculate its *earliest first location*, *latest last location*, *average location* and *average first location* for tokens of this word across all documents in the input. In addition, features are set to indicate whether a word appears in the first sentence and the number of times it appears in the first sentence, across all documents in the input. There are six features in this class, all of them are highly significant, ranked within the top 250 by WRS test ($p < 10^{-72}$).

Note that words close to the beginning of an article may not be that important for other domains, such as novels and legal texts (Ceylan et al., 2010). In fact, in contrast to news articles, materials that appear later in meeting transcripts (Murray et al., 2006) and lectures (Beigman Klebanov et al., 2014) tend to be more important.

3.5.2 Word Properties

We discuss three kinds of word properties: part-of-speech, named entity categories, and word capitalization.

Part-of-speech: Part-of-speech (POS) is useful in improving the identification of keyphrases that are used for indexing (Hulth, 2003). In summarization, the POS information is also effective in eliminating unimportant content (e.g., lead adverbials, gerund clause) to make the summaries more concise (Dunlavy et al., 2003; Conroy et al., 2006b). One relevant analysis appears in Gillick (2011), who studied the distribution of POS tags in the document sets and the summaries. Recently, Woodsend and Lapata (2012) use POS information as features for estimating word importance. The estimation result is used as an indicator that helps to decide whether or not a node in a parse tree should be removed in a summarization system that uses compressed sentences of the input. In general, however, POS tags are not often used as features for estimating word importance in summarization.

Features	**Sample words**	**prop**	**WRS**	**+/-**	r_f
NNP	president, November, Clinton	NA	3.7e-24	+	15.9%
NNPS	States, Nations, embassies	NA	1.2e-18	+	34.9%
VBN	accused, killed, held, arrested	NA	3.3e-6	+	14.5%
VBG	taking, adding, including, speaking	NA	2.7e-5	-	7.6%
RB	ago, recently, long, apparently	NA	3.1e-5	-	6.4%
NNS	years, officials, countries, victims	NA	4e-5	+	13.1%
FW	el, hage, cardinal, es	NA	7.3e-5	+	33.3%
VB	make, give, carry, put, face	NA	0.0003	-	7.8%
VBZ	appears, remains, includes, means	NA	0.0003	-	7.8%
CD	13, 1998, million, billion	NA	0.002	+	12.5%
VBP	remain, include, feel, fear	NA	0.004	-	6.2%
JJR	larger, lower, higher, greater	NA	0.007	-	6.2%
VBD	killed, began, voted, reported	NA	0.011	+	13.1%
NN	president, government, state	NA	0.040	+	11.8%
Organization	international, state, national	NA	3.6e-61	+	25.5%
Location	U.S., States, United, York, Saudi	NA	4.2e-44	+	22.4%
Other entities	president, time, international	NA	8.2e-17	+	11.0%
Person Names	Ms., David, John, Ali, Michael	NA	0.009	-	9.5%
Date	November, October, 1998, years	NA	0.023	+	12.3%
Money	million, billion, 13, 30	NA	0.035	+	13.4%
Ever capitalized?	Bush, United, British, China	3e-35	2.2e-35	+	16.0%
Capitalization ratio?	NA	NA	2.1e-31	+	NA
All capitalized?	Bush, United, British, China	2.6e-11	2.1e-11	+	13.8%

Table 3.3: The significant part-of-speech, named entity and capitalization features. We show their p-values by Wilcoxon rank-sum test (WRS). For binary features, we also show their p-values by proportion test. +/- indicates more in the summary/input. r_f indicates the percentage of words with this feature tag in the input that are included in human summaries; the mean r_f of all words is 10.9%.

In our work, we include POS tags for each individual word. Here we use the Stanford POS-Tagger (Toutanova et al., 2003). We have one real-valued feature corresponded to each POS tag: let N_w denote the number of occurrences of word w in the input and let N'_w denote the number of occurrences of word w with POS tag t in the input, the value of this feature is equal to N'_w/N_w. In most cases only one feature gets a non-zero value.

Of all POS tags, 14 of them are significant (see Table 3.3). There are more nouns (NNS, NNPS, NN), numbers (CD) and past tense verbs (VBN, VBD) in the summaries compared to the input. There are fewer present tense verbs (VB, VBG, VBP, VBZ), comparative adjectives (JJR) and adverbs (RB) used in summaries. For a description of the part-of-speech tag sets, see Santorini (1990).

We also quantify the percentage of words with certain POS-tags in the input that are also included in human summaries. Formally, let W_I denote the set of content words in the input I and let S_I denote the set of content words in the summary corresponded to I. Let $W_{I,f}$ and $S_{I,f}$ denote the set of words in W_I and S_I that have the POS tag corresponded to feature f, respectively. For each input I and feature f, we compute:

$$r_{I,f} = \frac{|W_{I,f} \cap S_{I,f}|}{|W_{I,f}|} \tag{3.5}$$

The final r_f that corresponds to feature f is equal to the mean of $r_{I,f}$ over all input sets. We show the r_f for all significant features in Table 3.3. Of all content words in the input, 10.9% of them appears in human summaries. The r_f of many significant features (e.g., NNPS, VBN) are much higher than 10.9%.

Named Entities: Named entity recognition (NER) classifies words into predefined categories (e.g., date, time, organization). For each word, we include its named entity label, derived by the Stanford Name Entity Recognizer (Finkel et al., 2005). Among the eight NE features, six of them are significant: there are more *Organization*, *Location*, *Other entities*, *Date*, *Money*; less *Person Names* in human summaries (see Table 3.3). Indeed, many of the words in the *Organization* (e.g., international, state, national) and *Location* categories (e.g., States, Saudi, river) are related to the most critical events of the input. The significantly less occurrences of *Person Names* might be because abstractors would only select the most important names (e.g., Bush), while the large number of other names (e.g., David, John, Michael) that appear in the input documents are left out.

Capitalization: For each word, features are set for whether or not this word has been capitalized, the ratio of its capitalized occurrences, and whether or not all its occurrences are capitalized. Sentence initial words are excluded before computing. Capitalized words are more likely to appear in summaries, as shown in Table 3.3.

3.5.3 Estimating Word Importance from Summaries

Prior work has shown that having estimates of sentence importance can help in estimating word importance (Wan et al., 2007; Liu et al., 2011; Wei et al., 2008). Similarly, one can assume that whether or not one word appears in a competitive machine-generated summary can also be a strong indicator of word importance. In our work, we use a Kullback-Leibler (KL) divergence (Greedy-KL) based summarizer that achieves very competitive performance (see Table 4.5). After the summaries have been generated, features are set to indicate whether the word appears and the number of times the word appears in the Greedy-KL summary. Both features are highly significant, ranked within the top 100 ($p < 10^{-300}$, WRS test).

Below we introduce the Greedy-KL summarizer:

The Greedy-KL summarizer

The Greedy-KL summarizer (Haghighi and Vanderwende, 2009) aims at minimizing the KL divergence between the probability distribution of words in the summary and that of the input. This summarizer is a component of the popular topic model approaches (Daumé III and Marcu, 2006; Celikyilmaz and Hakkani-Tür, 2011; Mason and Charniak, 2011) and achieves competitive performance with minimal differences compared to a full-blown topic model system. Specifically, let P, Q denote the unigram distribution of the summary (S) and that of the input. This summarizer minimizes the following formula:

$$KL(P \parallel Q) = \sum_{w \in S} P(w) \cdot \ln \frac{P(w)}{Q(w)} \tag{3.6}$$

Algorithm 2 Greedy-KL Summarizer

```
procedure Greedy-KL(I, L)
    $p$ denotes the word distribution of a document set.
    while $Len(Sum) < L$ do
        j = $argmin_i\ KL(p(Sum \bigcup s_i) \parallel p(I))$        $s_i$ are input sentences in $I$
        $Sum \leftarrow Sum \bigcup \{s_j\}$
    end while
    return $Sum$
end procedure
```

Note that the formula we use here is different from Haghighi and Vanderwende (2009), as we minimize $KL(P \parallel Q)$ instead of $KL(Q \parallel P)$. Indeed, if one optimizes $KL(Q \parallel P)$, then smoothing has to be performed to P. Because otherwise, $P(w) = 0$ and $Q(w) \neq 0$ would cause $KL(Q \parallel P)$ undefined. However, if one optimizes $KL(P \parallel Q)$, since the words in P always appear in Q, $Q(w) = 0$ and $P(w) \neq 0$ will not happen. In this way, we do not need to perform smoothing.[10]

However, minimizing $KL(P \parallel Q)$ is computationally expensive. Indeed, let M, N denote the number of sentences in the input and the summary respectively, one needs to enumerate $O(M^N)$ possibilities. Therefore, a greedy algorithm (Greedy-KL) is presented, which iteratively selects a sentence that minimizes $KL(C \parallel Q)$, where C is the word distribution of the current summary. Let I denote the input and let L denote the length limit, let the function $Len(t)$ denote the length of text t. Pseudo code of the Greedy-KL summarizer is shown in Algorithm 2.

3.5.4 Context Features

We use context features here, based on the belief that the context around a word affects the importance of this word. This idea has been shown useful for keyphrase identification (Mihalcea and Tarau, 2004). For the context here, we simply consider the words before and after the target word.

[10] Moreover, the summarizer that optimizes $KL(P \parallel Q)$ has a better performance than optimizing $KL(Q \parallel P)$.

Formally, let w denote the target word. Let L_w and R_w denote the content words before and after w in the input, respectively. Let $w.f_i$ denote a feature for w, the newly extended word-before features $w.l_{f_i}$ can be computed as:

$$w.l_{f_i} = \sum_i p(w_l) \cdot w_l.f_i, \ \forall w_l \in L_w \tag{3.7}$$

Here $p(w_l)$ is the probability w_l appears before w. The word-after features $w.r_{f_i}$ are computed in the same way. Here we only extend the most basic features, including word probability, LLR and MRW scores, location, capitalization, and feedback from KL-divergence summaries.

For each w, we also compute the weighted average ranks of the words in L_w and R_w, according to their ranks by word probability, LLR and MRW. For w, the weighted average rank $w.l_{rank_M}$ for its left context is:

$$w.l_{rank_M} = \sum_i p(w_l) \cdot w_l.rank_M, \ \forall w_l \in L_w \tag{3.8}$$

,where M is the current ranking method, $w.rank_M$ indicates the rank of a word using this method. If L_w is $\emptyset$, we assign a very large value to $w.l_{rank_M}$. Features are then set based on the weighted average rank: a feature is set to 1 if the rank is smaller than k, otherwise 0. Here k are preset cutoffs, which are the same as what we used in Section 3.5.1. All 220 features in this class are significant, among which 216 are highly significant ($p < 10^{-3}$, WRS test).

3.5.5 Unigrams

This is a binary feature which corresponds to each content word that appears at least twice in the training data. This feature helps to learn the specific words from the input that tend to be included in (or excluded from) human summaries. Among all 8049 unigram features, 199 are significant by proportion test, 651 are significant by WRS test. Despite the high number of significant features, the predictive power of these features are not as high as the more general ones.

Unigrams	**prop**	**WRS**	**t**	+/-	Unigrams	**prop**	**WRS**	**t**	+/-
president	6e-23	2e-24	6e-5	+	time	0.177	0.093	3e-7	-
international	4e-16	2e-17	5e-4	+	week	0.177	0.093	3e-7	-
government	3e-11	5e-12	1e-3	+	Wednesday	0.190	0.099	6e-7	-
killed	3e-11	2e-12	2e-3	+	Monday	0.204	0.106	1e-6	-
accused	5e-11	2e-12	2e-3	+	told	0.219	0.114	2e-6	-
people	3e-10	3e-7	7e-3	+	statement	0.235	0.122	3e-6	-
November	2e-9	7e-12	0.010	+	Tuesday	0.253	0.131	5e-6	-
years	2e-7	4e-6	0.012	+	Saturday	0.253	0.131	5e-6	-
killing	2e-7	2e-7	0.014	+	put	0.272	0.141	8e-6	-
large	1e-6	8e-7	0.015	+	spokesman	0.272	0.141	8e-6	-

Table 3.4: Top 10 most significant summary-biased (+) and input-biased (-) words based on proportion test (prop). We also show their p-values by Wilcoxon rank-sum test (WRS) and t-test (t).

Table 3.4 shows the top 10 summary-biased and input-biased words by proportion test. The summary-biased words have much smaller p-values, while none of the input-biased words are identified as significant (the result is similar for WRS test). To investigate why proportion test gives such a result, we look into two examples: the summary-biased word "*president*" and the input-biased word "*time*". Among $3,234$ training instances that are labeled as one, 18 of them correspond to "*president*", the ratio is 5.6×10^{-3}; among $29,789$ instances that are labeled as zero, 8 of them correspond to "*president*", the ratio is 2.6×10^{-4}. According to proportion test, these two ratios are significantly different. The word "time" appears 29 times in the input and 0 times in the summary. Among the instances that are labeled as one and zero, the ratios are 0 and 9.7×10^{-4} respectively, which is judged to be insignificant by proportion test. From the example of "time", we can see that none of the input-biased features are significant.

While studying the predictive power of features, it is important to *choose an appropriate test*. In fact, we have used t-test in our early experiments, where many spurious words are judged as significant (see Table 3.4). Among proportion test, WSR test, and t-test; proportion test gives the most conservative result, with the smallest number of features identified as significant.

The main disadvantage of unigram features is that they are sparse. The predictive power of these features are weak, and they might carry noisy information specific to the training data. Below, we briefly describe two classes of features that capture the global importance of categories (Section 3.5.6) or words (Section 3.5.7). A detailed exploration of global knowledge can be found in Chapter 5.

3.5.6 Dictionary Features: MPQA and LIWC

One way to deal with the sparsity of unigrams is to cluster these unigrams based on their properties. Here we describe features based on two hand-crafted dictionaries that group words into salient subjectivity, semantic and lexical categories. The value of the feature is 1 for a word if it appears in the category that this feature corresponds to. Specifically, we employ Multi-Perspective Question Answering (MPQA) (Wiebe and Cardie, 2005) for subjectivity analysis and Linguistic Inquiry and Word Count (LIWC) (Tausczik and Pennebaker, 2007) for topic analysis.

Since we will present an detail analysis on the features derived from MPQA and LIWC in Chapter 5, here we only describe the main findings.

The MPQA dictionary contains words labeled with three different polarities (positive, neutral, negative) and two different subjectivities (strongly subjective, weakly subjective). The combinations correspond to six features. It turns out that words with strong subjectivity—whether with positive, neutral or negative polarity—are less likely to be included in the summaries. Here we can have two explanations: (1) the main topic of an input tends not to be too subjective, (2) abstractors tend not to be too subjective while summarizing articles. We will show in Section 5.3.1 that the latter explanation is more plausible. There is no significant difference on weak subjectivity categories.

Another dictionary that we use is LIWC, which focuses on lexical and semantic properties of the words. Of all 64 LIWC features, 16 are significant by proportion test. Interesting categories that appear at a higher rate in human summaries include death, anger, achievements, money and negative emotions. The words that appear at a lower rate in human summaries include function words, words with present tense, as well as words that belong to the semantic categories of hear, body, friends, and positive emotions.

3.5.7 Intrinsic Importance of Words (Global Indicators)

Some words are of intrinsic importance to people, regardless of the input. In Section 5.4, we will describe how such intrinsic word importance scores are estimated based on the summary-article pairs from a large corpus. In particular, we call such importance scores *global indicators*, in contrast to traditional methods that estimate word importance based on the local input. In this section, we simply focus on how the feature vectors are constructed, assuming that we have the global indicators of words already.

Let L denote a list of words, ranked by these intrinsic scores (global indictors). For each word, features are set to determine whether or not it is within top-k or bottom-k in L (one feature for top-k, one feature for bottom-k), where k is selected from a set of pre-defined values.[11] These features indicate whether or not people consider the word to be among the top k most (least) important ones, according to the metric of ranking word importance. In our work, five methods are used to rank word importance, which corresponds to five lists.

There are 70 binary features in this category, of which 52 are significant (proportion test). The predictive power of these features is very high, with 48 of them satisfying $p < 0.001$.[12]

[11] 100, 200, 500, 1000, 2000, 5000, 10000 in this case.

[12] The result of WRS test is very similar: 53 features are significant, 49 features have $p < 0.001$.

3.6 Experiments

Our experiments are conducted on the DUC 2003, 2004 datasets, the last two DUC/TAC datasets that focus on generic multi-document summarization. DUC 2003 is used for training, DUC 2004 is used for testing, a conventional setting used in many prior papers on summarization (Yih et al., 2007; Takamura and Okumura, 2009; Kulesza and Taskar, 2012). We assign a training label for each unique content word in each of the input sets. The label is equal to 1 if the word appears in any human summaries of the input, otherwise it is equal to 0.

To train our model, we use L_1-regularized (binary) logistic regression. We use the model implemented in Liblinear (Fan et al., 2008) with default parameter settings.[13] Formally, let $\{(\mathbf{x_1}, y_1), \ldots (\mathbf{x_m}, y_m)\}$ denote the training data, where is $\mathbf{x_i} \in \mathbb{R}^N$ is an N-dimensional feature vector, y_i is a class label. Let $\boldsymbol{\theta} \in \mathbb{R}^N$ denote the parameters. Logistic regression models the probability distribution of y given $\mathbf{x}$ as below:

$$p(y = 1|\mathbf{x}; \boldsymbol{\theta}) = \frac{1}{1 + exp(-\boldsymbol{\theta}^T \mathbf{x})} \tag{3.9}$$

For regularized logistic regression, the goal is to find parameters $\boldsymbol{\theta}$ which would solve the following optimization problem (Ng, 2004):

$$argmax_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p(y_i|\mathbf{x_i}; \boldsymbol{\theta}) - \alpha \cdot R(\boldsymbol{\theta}) \tag{3.10}$$

Here $R(\boldsymbol{\theta})$ is the regularization term. For L_1 regularized logistic regression, we have $R(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\| = \sum_{i=1}^{n} |\theta_i|$. α is a tradeoff between overfitting and underfitting the data.

It has been frequently observed that L_1 regularization causes many parameters in $\boldsymbol{\theta}$ to be equal to zero (Ng, 2004). This means, L_1 regularization is helpful to select features, where we believe that many of the features can be ignored (e.g., many of the unigram features in our case). Another popular method is L_2-regularized logistic regression, which does not have the functionality of feature selection.[14] Thus we use

[13] ./train -s 6

[14] If we train our model using L_2 regularized regression, $8,543$ features (parameters) are active.

L_1-regularized logistic regression. Among all $8,671$ features in our model, 809 of them are active (selected).[15] Regularization, in general, helps to prevent overfitting.

After $\boldsymbol{\theta}$ has been estimated, we compute $p(y = 1|\mathbf{x}; \boldsymbol{\theta})$ while predicting. The output of our model indicates the probability that one word appears in human summaries, which we regard as the estimates of word importance. The performance of our model is evaluated in identifying summary keywords on the DUC 2004 data. The predicted keywords are the top k words with the highest importance scores.

3.6.1 Keyword Identification

We compare our logistic regression model (RegSum) with three unsupervised word weighting approaches: word probability (Prob), log-likelihood ratio (LLR) and Markov random walk model (MRW). To show the effectiveness of new features, we compare our results with a logistic regression model (RegBasic) trained only on frequency (Prob, LLR, MRW and document frequency) and location related features (described in Section 3.5.1). These features resemble the ones used for ranking the importance of words in recent summarization work (Yih et al., 2007; Takamura and Okumura, 2009; Sipos et al., 2012).

We first evaluate the performance of different systems using precision (P), recall (R) and F_1-measure (F). The final P, R and F are the average P, R, F over all input sets. Figure 3.3 shows the performance of systems when selecting the top 100 words with the highest weights as keywords. Each word from the input that appeared in any of the four human summaries is considered as a gold-standard keyword (G_1).

Among unsupervised approaches, Prob identifies keywords better than LLR and MRW by at least 0.04 on F_1-score. We are surprised to find that RegBasic does not outperform Prob, even though it includes location information. However, we will show in Section 4.1.3 that a summarizer based on the weights predicted by

[15] However, L_1 regularization does not tackle collinearity. Many correlated parameters are active and assigned weights.

RegBasic produces much better summaries compared to summarizers whose weights are estimated by unsupervised methods. Our system (RegSum) gives 0.025 F_1-score improvement over Prob and RegBasic, more improvement over the other approaches. All improvements are statistically significant ($p < 0.05$) by two-sided Wilcoxon signed-rank test.

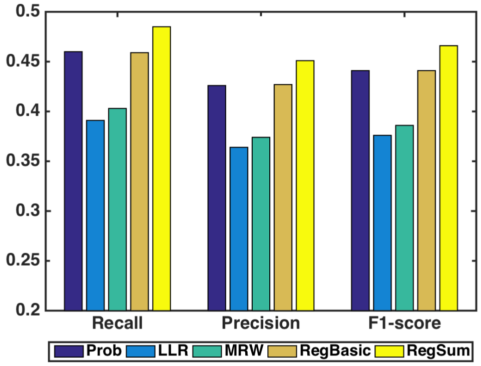


Figure 3.3: Precision, Recall and F_1-score of keyword identification, 100 words are selected, G_1 is used as the gold-standard.

Table 3.5 shows the F_1-score of keyword identification using different G_i as gold-standards. Ranking words by their probabilities in the input performs well in identifying words that appear in all human summaries. On the other hand, our model achieves a significant improvement in predicting words in G_1 or G_2 (words used in at least one or two human summaries). Indeed, the words that are used in only one or two human summaries are the ones that are difficult to be predicted based on frequency of a word in the input.

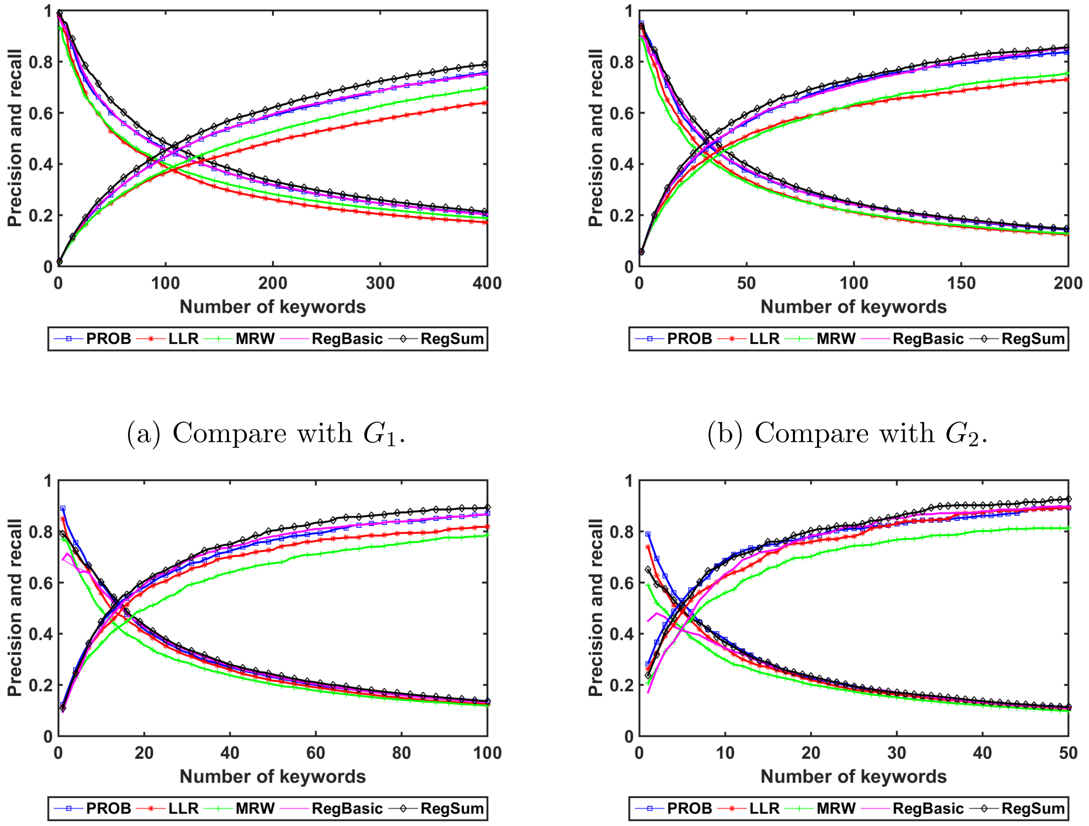
We plot the precision, recall and F_1-score, changing with different number of words selected (Figure 3.4, Figure 3.5). Here we use different G_i as gold standards. Among the three unsupervised approaches, Prob performs the best. LLR is a poor method in identifying words that appear in at least one summary (G_1), while MRW is a poor method in identifying words that appear in all human summaries (G_4). Our full blown model (RegSum) performs significantly better than all other approaches

G_i	#words	Prob	LLR	MRW	RegBasic	RegSum
G_1	120	0.444	0.372	0.387	0.443	**0.469**
G_1	100	0.441	0.376	0.387	0.441	**0.466**
G_1	80	0.431	0.373	0.379	0.429	**0.456**
G_2	40	0.464	0.423	0.409	0.465	**0.491**
G_2	35	0.465	0.428	0.410	0.475	**0.503**
G_2	30	0.469	0.428	0.409	0.477	**0.499**
G_3	20	0.486	0.466	0.414	0.491	**0.502**
G_3	15	0.507	0.468	0.418	0.496	**0.518**
G_3	10	**0.491**	0.458	0.414	0.468	**0.491**
G_4	7	**0.500**	0.467	0.397	0.416	0.477
G_4	6	**0.500**	0.466	0.404	0.398	0.479
G_4	5	**0.497**	0.450	0.387	0.379	0.471

Table 3.5: Keyword identification F_1-score with different number of words selected. **Bold** indicates that one method performs the best among the five methods.

when G_1 and G_2 are used as the gold standards. However, it performs inferior to Prob when identifying words that appear in all human summaries, as reflected by the peak performance (F_1-score) in Figure 3.5 (d). RegBasic has a similar performance compared to Prob when G_1, G_2 and G_3 are used as the gold standards, but performs poorly in identifying words that appear in all human summaries (Figure 3.4 (d)).

The two supervised methods (RegBasic and RegSum) do not give a good performance in identifying the words used in all human summaries (G_4). Here we try to explain why this happens. While training the logistic regression model, we assign label 1 to all words that are used in human summaries. We do not differentiate between the words that appear in k human summaries. Therefore, the weights predicted is the probability of one word used in human summaries, not the number of human summaries one word appears in. As a result, the weights assigned to the top few words are somewhat mixed up (e.g., the word "Sen" in Table 4.2), though these words are all likely to be used by human. This problem is more severe for RegBasic,



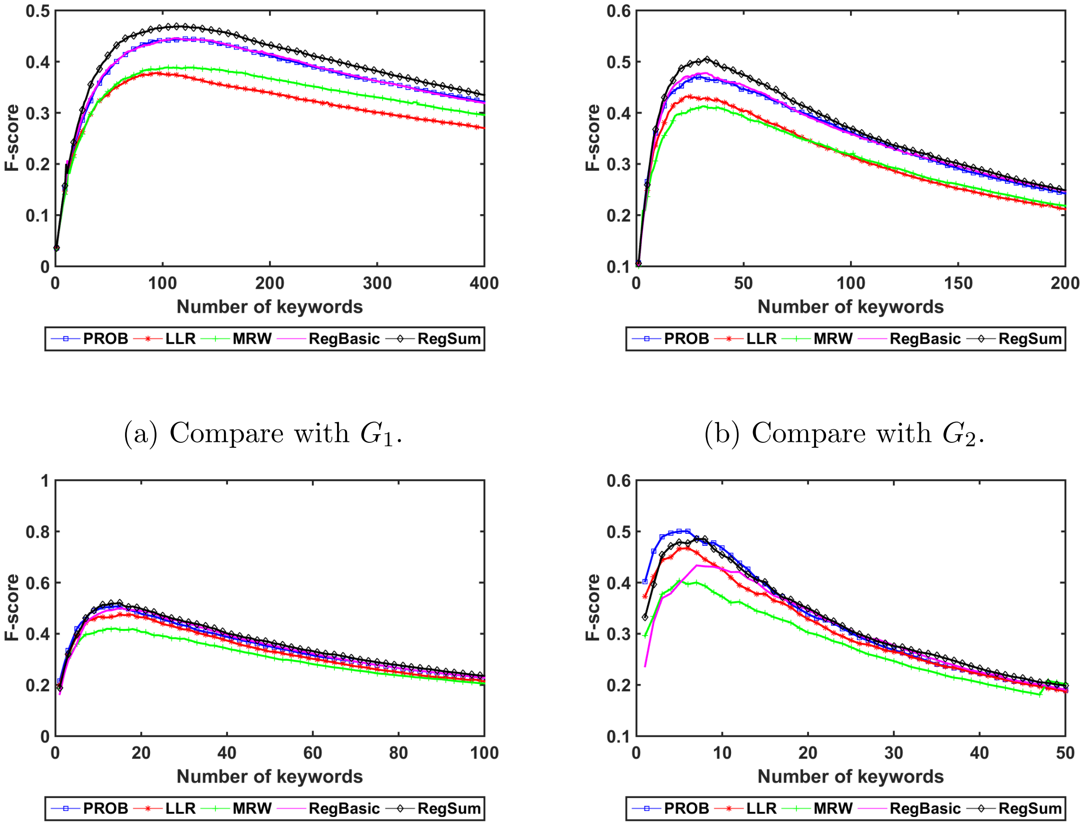
(a) Compare with G_1.

(b) Compare with G_2.

(c) Compare with G_3.

(d) Compare with G_4.

Figure 3.4: Precision and recall derived by comparing the selected words to the gold-standard keywords used in at least i human summaries (G_i). The increasing functions are recall. x-axis is the number of keywords selected.



(a) Compare with G_1.

(b) Compare with G_2.

(c) Compare with G_3.

(d) Compare with G_4.

Figure 3.5: F_1-score derived by comparing the selected words to the gold-standard keywords used in at least i human summaries (G_i). x-axis is the number of keywords selected.

where the number of features is small. We suspect that this problem might be alleviated if the number of human summaries that one word appears in are used as the training labels. For this case, we need to use other regression algorithms, as binary logistic regression assumes the training labels to be 0 or 1.

3.6.2 The Keyword Pyramid Method

Our gold-standard word lists G_1, G_2 and G_3 include words that appear in different number of human summaries. Apparently, words that are used in more human summaries should be assigned higher importance weights during evaluation. However, this is not reflected if we evaluate the performance by precision, recall and F_1-score. To address the problem, we present a weighted approach to evaluate the identification of summary keywords. Moreover, our approach generates a single score as its final result, which makes the evaluation simpler. We call our method the *Keyword Pyramid Method*, as it is inspired by the Pyramid Method (Nenkova et al., 2007).This method works as follows:

Consider an input set with n human summaries. For each word w, we assign a weight $t(w) = i \ \ (0 \leq i \leq n)$, representing the number of summaries that includes w. This process resembles the step of making the pyramid in the Pyramid Method.

Consider we are then given a list of words L returned by a word weighting method. Let w_i denote the word ranked the ith in L and let L_k denote the top k words in the list L. The total weights of all words in L_k can be written as:

$$Coverage(L_k) = \sum_{i=1}^{k} t(w_i) \tag{3.11}$$

Similar to the Pyramid Method, we normalize $Coverage(L_k)$ by the optimal weight that can be achieved with k words selected. The optimal score is computed from a list where all words that appear n times are ranked first, followed by all words that appear $n - 1$ times, etc. Let Opt_k denote the optimal coverage weight of the

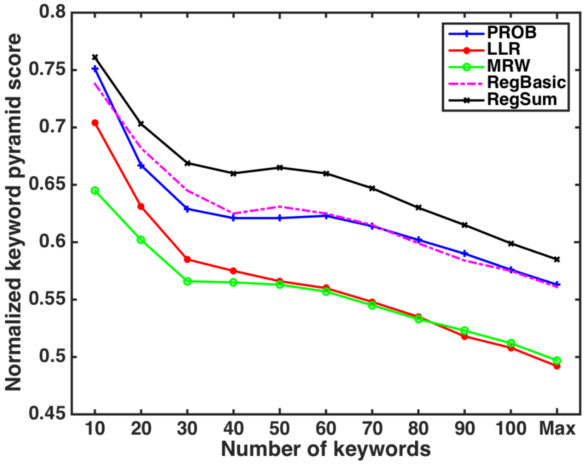


Figure 3.6: The Keyword Pyramid Scores of five methods.

first k words, the Keyword Pyramid Score $Pyr(L_k)$ is:

$$Pyr(L_k) = Coverage(L_k)/Opt_k \tag{3.12}$$

We now compare the performance of the five systems. Figure 3.6 shows how the final score changes with k, where we take the average of $Pyr(L_k)$ across all input sets. When k is equal to ten, our system (RegSum) performs similar to ranking words based on probability (Prob). RegSum performs significantly better than all other methods when $k \geq 20$. Among the other methods, RegBasic gives a slightly better performance than Prob. LLR and MRW do not perform as well as the other approaches. In summary, we have introduced an easy and straightforward method for evaluating the performance of identifying words used in human summaries.

3.6.3 Effectiveness of Features

Below we study the contribution of different features by two experiments.

Feature Ablation Experiment

First, we conduct an ablation experiment, which leaves out one class of features at a time from RegSum. The performance is reported in Table 3.6. Here we report the

	Keyword Pyramid Score			k words vs G_i (F_1-score)			
	k=100	k=50	k=10	100 vs G_1	35 vs G_2	15 vs G_3	6 vs G_4
RegSum	.601	.665	.761	.466	.503	.518	.479
-frequency (freq)	.578	.621	.701	.449	.445	.466	.464
-location (loc)	.588	.642	.749	.456	.487	.496	.436
-freq+loc	.491	.539	.596	.374	.391	.383	.363
-MPQA	.601	.664	.763	.467	.500	.524	.484
-LIWC	.598	.664	.761	.461	.498	.522	.473
-NYT	.599	.662	.757	.463	.500	.521	.456
-all global	.601	.663	.755	.463	.494	.525	.460
-property	.585	.630	.735	.452	.468	.488	.445
-KL	.600	.664	.764	.464	.497	.523	.478
-unigrams	.603	.662	.757	.466	.493	.517	.478
-context	.603	.667	.760	.469	.505	.510	.464

Table 3.6: Performance of ablating different feature classes. We report the Keyword Pyramid Score and F_1-score that compares the top k words to different G_i.

Keyword Pyramid Score achieved with k words selected ($k = 10, 50, 100$) as well as the performance of identifying words used in i human summaries (denoted as k vs G_i in Table 3.6, where k indicates the number of keywords identified).

We first look into the most basic feature sets: frequency and location related features. These two are widely acknowledged as strong indicators of word importance. Ablating frequency related features leads to a large decrease in performance, especially when G_2 and G_3 are used as the gold standards. Ablating location related features also leads to a noticeable decrease, though not as large as word frequency (except for 6 vs G_4). Interestingly, ablating both frequency and location based features leads to a drastic decrease in performance. Therefore, removing either class of features does not affect the performance to a great extent, while removing both greatly affects the performance.

We next ablate the features related to global knowledge. Removing the global indicator features (NYT) leads to a decrease in performance in general, except for 15 vs G_3. We observe that ablating the LIWC features only results in a small decrease,

	Keyword Pyramid Score			k words vs G_i (F_1-score)			
	k=100	k=50	k=10	100 vs G_1	35 vs G_2	15 vs G_3	6 vs G_4
RegBasic (freq+loc)	.575	.631	.738	.441	.475	.496	.398
+property	.595	.650	.762	.458	.492	.531	.458
+KL	.574	.632	.735	.438	.471	.501	.421
+unigrams	.581	.642	.724	.447	.487	.485	.405
+context	.571	.621	.723	.435	.465	.495	.411
+MPQA	.573	.628	.737	.438	.472	.494	.404
+LIWC	.578	.637	.734	.441	.476	.502	.390
+NYT	.586	.636	.737	.452	.473	.497	.429
+all global	.582	.634	.730	.448	.476	.493	.430

Table 3.7: Performance of including different feature classes. We report the Keyword Pyramid Score and F_1-score that compares the top k words to different G_i.

while ablating the MPQA features even leads to an improvement. Finally, we ablate all global knowledge features, which decreases the performance most of the times (except for $k = 100$ and 15 vs G_3). In general, the global knowledge related features bring a small improvement in identifying summary keywords. Though, we will show in Section 5.5 that the global indicators are helpful in identifying low frequency words in the input that are used in human summaries.

We then investigate the feature classes apart from frequency, location and global knowledge. The word property (*property*) features include part-of-speech tags, named entity tags and capitalization information. These features are useful, with removing which leads to a noticeable decrease in performance on all metrics. The features (*KL*) that indicate whether one word has appeared in a Greedy-KL summary are among the most significant ones. However, removing them does not affect the performance much. We suspect the reason is because these features are in fact related to the frequency based ones, as the Greedy-KL summarizer takes advantage of the probability of a word in the input. The change in performance after ablating unigram features is also small (except for 35 vs G_2). Ablating features related to context even improves the performance.

Feature Inclusion Experiment

We include one class of features at a time in addition to the frequency and location features (RegBasic) (Table 3.7). Among all feature classes, word property is the most useful. The next useful feature class is the global indicator class, where we observe an noticeable improvement for identifying the words in G_1/G_4 after these features are included. The third useful features are unigrams. The two feature classes (LIWC & MPQA) related to dictionary knowledge are not that effective. Similar to what we observe in Table 3.6, including context features decreases the final performance. Overall, none of the models in Table 3.7 perform as good as RegSum, which indicates that it is useful to include a rich variety of features.

3.7 Conclusion

In this chapter, we present a superior supervised model that identifies words used in human summaries. We explore features such as part-of-speech (POS), named entity (NE) tags, subjectivity, topic categories and global indicators. In terms of POS, we observe that human summaries include more nouns and past tense verbs, fewer comparative adjectives and adverbs. In terms of NE tags, there are more words that belong to *Organization* and *Location* types while surprisingly less words that belong to *Person Names* type in human summaries. Moreover, we observe that humans tend not to include too subjective words. We also observe that the features which quantify the intrinsic importance of words are among the most significant ones.

We have studied the contribution of different features. We show that frequency, location and word properties are the most important features. The global indicators derived from the NYT corpus also carry useful information. However, the effects brought about by ablating unigrams and greedy-KL features is limited. Ablating MPQA (word subjectivity information) and context information even improves the performance. In summary, though our analyses of features have revealed interesting findings, some significant features are not that effective.

Improving the estimation of bigram importance is also useful in improving summarizers based on Integer Linear Programming (ILP) (Li et al., 2013b). There, the summarizer optimizes the sum of bigram weights, where the weights are estimated by a supervised model. The authors show that using learned weights is better than using document frequency. Different from us, their model is evaluated on the datasets towards query-focused summarization (TAC 2008–TAC 2011). The features they used are also different from ours, which includes term frequency, overlap with the query and information about the sentences that include the bigram. In a later work of Li et al. (2015), the authors include a large variety of features mined from external resources to weigh bigrams, where our global importance feature is among one of them. Section 5.2 includes a more detailed review of this paper.

For future work, many of our proposed features—especially the word property and global knowledge ones—can be extended for estimating the importance of concepts other than unigrams and bigrams (e.g., named entities, phrases, syntactic subtrees). This might lead to an improvement for more sophisticated summarization systems. Future research directions also include employing the estimated word weights into compressive summarization systems, where the words to be removed can be decided based on these weights. Moreover, better estimation of word importance is also likely to improve systems for sentence compression that use only crude frequency related measures to decide which words should be deleted.

We will employ the estimates of word importance to generate summaries in Chapter 4. We will provide a more detailed discussion concerning features related to global knowledge in Chapter 5.

Chapter 4

Comparing Summarization Systems: Modular Comparison and Output Comparison

This chapter[1] focuses on comparing different generic summarization systems.

If one asks the question "which system produces the best summary", it is a relatively easy question to answer (at least seemingly). However, if one asks "why is System A better than System B", it is much more difficult to answer. Indeed, most summarization systems have different designs for each component of the summarization process. Therefore, one cannot answer the question like: "System A is better because it has a better preprocessing component." One can only say: "I don't know, but System A has a better performance."

The first half of this chapter aims to answer the more difficult question (Section 4.1). Specifically, based on a modular greedy summarization system, we compare different word weighting approaches.

[1] Section 4.1 is adapted from Hong and Nenkova (2014a) and Section 4.2 is adapted from Hong et al. (2014).

Moreover, the seemingly easier question “which system produces the best summary” is also difficult to answer in practice. Different systems were often evaluated on different datasets, with different evaluation metrics. For example, the six state-of-the-art systems we discussed in Section 4.2 were originally evaluated on four different datasets and evaluation metric combinations.

The second half of the chapter aims to tackle this difficulty (Section 4.2). Specifically, we present a repository that includes the summaries from multiple state-of-the-art and popular baseline systems on a common set. We evaluate the performance of those systems using ROUGE settings that have high correlations with manual evaluations, and also conduct paired statistical significance test to compare systems. Our repository also makes it feasible for later work to compare the summaries from their systems to that from the systems in our repository.

4.1 Modular Comparison

4.1.1 Introduction

Summarization systems can be regarded as a pipeline that includes multiple components (e.g., pre-processing, word weighting, sentence weighting, summary generation). Different researchers have different designs for each component, which makes it difficult to study the real contribution of each component towards the final content quality. In this section, we tackle this problem by conducting modular comparisons: we only change the methods used in one component at a time, a design that makes direct comparison possible. Here we decompose a summarization system into two components: a word weighting component and summary generation component.

Formally, let I denote the input and let f denote the summarization process. Let f_1 and f_2 denote the word weighting method and summary generation method, respectively. The procedure of producing summary S can then be represented as:

$$S = f(I) = f_2(f_1(I), I) \tag{4.1}$$

In the thesis, we study the effect of different f_1 towards the final summary quality. To do this, we present a modular greedy generic summarization system as our f (GreedySum): a system that is simple, fast and transparent (Section 4.1.2). Then we present a class of empirical studies which compare different f_1 (Section 4.1.3). Specifically, we aim to answer the following three questions:

- Which f_1 can lead to the best summary quality?
- Will better estimation of word importance lead to better summaries?
- Originally, the words in the input are weighted by their importance, which are real values. However, it has been shown in prior work that for log-likelihood ratio weighting (Section 3.4.2) in particular, weighing words using binary values (i.e., assign weight 1 to the keywords and weight 0 to the others) is better than using real values (Gupta et al., 2007). Can this idea be extended successfully to other word weighting methods?

To answer the first and the second question, we compare the methods presented in Chapter 3. Since it is difficult to evaluate different estimates of word importance, we use the evaluation result of summary keyword identification as a proxy (described in Chapter 3). We show that the best model in identifying summary keywords also performs the best for summarization, which produces summaries comparable to the state-of-the-art. A simple supervised model that uses only frequency and location information can also produce summaries of high quality. This answers the first question. We then show that a model that can better identify summary keywords does not always lead to better summaries. Thus the answer for the second question is: most of the times, but not always.

To answer the third question, for each $f_1(w)$ whose output is a real value, we form a class of comparison groups $f_1'(w,k)$ by binary weighting. $f_1'(w,k)$ is equal to 1 if w is among the top k words ranked by $f_1(w)$, otherwise 0. The answer is: it depends. We empirically show that binary weighting works better if $f_1(w)$ is weighted by unsupervised methods, while real value weighting works better if $f_1(w)$ is learned using our proposed logistic regression model.

4.1.2 A Modular Greedy Summarization System

This section presents a modular greedy summarization system. This system is similar to the standard word probability baseline (Nenkova and Vanderwende, 2005; Nenkova et al., 2006; Nenkova, 2006), but we explore a range of methods for estimating word importance (e.g., the methods introduced in Section 3). The redundancy handling is also different from Nenkova and Vanderwende (2005) and Nenkova et al. (2006).

Let I denote the input and let $h(w)$ denote the weighting function of individual words. The $h(w)$ are the main differences between methods. Our summarizer (GreedySum) includes the following steps:

Step 1: For each content word $w \in I$, we employ $h(w)$ to estimate its importance. Since the functions $h(w)$ are undefined for stopwords (as discussed in Section 3.3), we have $h(w) = 0$ for those words, i.e., we consider stopwords to be uninformative.

Step 2: For each sentence $s_i \in I$, it is assigned an importance weight as follows:

$$Score(s_i) = \frac{\sum_{w_j \in s_i} h(w_j)}{len(s_i)} \tag{4.2}$$

Here $len(s_i)$ is the number of words in sentence s_i. This simple weighting method has been shown powerful in estimating sentence importance in prior work (Nenkova and Vanderwende, 2005; Nenkova et al., 2006).

Step 3: Sort all sentences in descending order according to their importance $Score(s_i)$.

Step 4: Pick the highest scoring sentence among all unselected *valid* (see below) sentences.

Step 5: Check if the desired summary length has been reached. If not, go to Step 4.

The *valid* condition in Step 4 is mainly concerned with sentence length and non-redundancy of the summary. Two conditions need to be met for one sentence to be judged as valid:

- For summarization systems, it is common to block sentences that are too short since they may harm the content quality (Erkan and Radev, 2004; Gillick et al., 2009; Li et al., 2013a). In our system, a sentence can be included only if it is at least nine words in length, a setting used in the famous MEAD system (Radev et al., 2004a).

- An ideal summary should include sentences that are both informative and diverse. In our system, we maintain non-redundancy based on its previous context in the summary. A sentence is regarded as non-redundant if it is not similar to any sentences already in the summary, measured by cosine similarity on binary vector representations with stopwords excluded. We use the cut-off of 0.4 for cosine similarity.[2] This value was tuned on the DUC 2003 dataset, by testing the impact of the cut-off value on the ROUGE-2 score for the final summary. Possible values ranged between 0.1 and 0.9, with a step of 0.1.

Below we discuss the advantages and disadvantages of our system:

Advantage 1: As pointed out in Nenkova et al. (2006), many summarization methods score sentences using supervised methods, where $h(w)$ is used as one of the features (Litvak et al., 2010; Ouyang et al., 2011; Wang et al., 2013). The use of these sophisticated yet non-transparent methods make it difficult to study the contribution of $h(w)$ towards sentence quality and final summary quality.

[2] Note this is different from Hong and Nenkova (2014a), where the cut-off was 0.5.

By using our method, the process of generating sentences becomes more transparent. This makes it easier to study the effect of $h(w)$ on $Score(s_i)$, which in turn makes it easier to study the effect on summary quality.

Advantage 2: It is easy to implement: the code is no more than 200 lines of Python.

Advantage 3: It is fast. Let M denote the number of sentences. Step 2 has time complexity $O(M)$ and step 3 has time complexity $O(M \cdot \log M)$. At step 4, we need to check redundancy with all sentences that are already in the summary. Let c denote the number of sentences in the output (a small number), the complexity is only $O(c^2)$. In practice, it takes no more than 0.5 seconds to summarize a document set.

Disadvantage 1: The way of handling redundancy is relatively coarse, which is based on a pre-defined threshold. Doing this might risk including sentences which already have a high degree of overlap content with the current summary, or excluding sentences which are non-redundant.

Disadvantage 2: This algorithm iteratively picks the most important sentence in a greedy fashion, following the classical maximal marginal relevance (MMR) style (Carbonell and Goldstein, 1998). However, researchers have argued that greedy based methods may not be the optimal choice for achieving a high content quality of the entire summary (McDonald, 2007; Gillick, 2011). McDonald (2007) has presented an example that shows this. Suppose we have a very long sentence that is highly informative but also includes noises. Greedy based systems are likely to include this sentence in the first place. However, doing this may not be as good as including a few short sentences that have similar content quality and include less unnecessary inforamtion. This would give more space to incorporate informative content.

A growing body of research has tackled this problem by doing *global inference*, which select a set of sentences to optimize the *overall content quality* of the summary approximately (Lin and Bilmes, 2010; Lin and Bilmes, 2011) or exactly (McDonald, 2007; Gillick et al., 2009). This is in contrast to greedy based methods that select the next sentence based on the sentence itself and the current summary. In addition to the theoretical advantages, global inference based methods have also been shown to perform better than greedy based methods that optimize the same (or a very similar) objective (McDonald, 2007; Gillick, 2011). However, we will show in Section 4.2 that global based algorithms do not outperform several greedy based methods on the DUC 2004 dataset (see Table 4.4, Table 4.5). In my opinion, the main question for global based methods is to find an optimal way of quantifying the overall content quality. Future research may investigate how this question can be better addressed.

4.1.3 Comparing Word Weighting Methods

Unsupervised weighting vs Supervised weighting

We compare three unsupervised and two supervised methods of weighting word importance in summarization, i.e., of computing $h(w)$. The three unsupervised methods employ word probability (Prob), the log-likelihood ratio (LLR) test and Markov random walk model (MRW) (see Section 3.4). The first supervised model (RegBasic) uses features that are derived from the three unsupervised weighting methods (Prob, LLR, MRW) and word locations (see Section 3.5.1). The second supervised model (RegSum) uses all features introduced in Section 3.5. The performance is evaluated on the DUC 2004 dataset.

Table 4.1 shows the performance of the five approaches in terms of ROUGE-1, -2, -4 (R-1, R-2, R-4). R-2 is regarded as the main metric, since it provides the best agreement with manual evaluations (Owczarzak et al., 2012). We also show the F_1-score of these methods in keyword identification, by comparing the top 100 keywords with all words that are used in human summaries.

	k words vs G_i (F_1-score)				Summarization		
System	100 vs G_1	35 vs G_2	15 vs G_3	6 vs G_4	R-1	R-2	R-4
Prob	.441	.465	.507	.500	0.3570	0.0826	0.0103
LLR	.376	.428	.468	.466	0.3467	0.0737	0.0078
MRW	.387	.410	.418	.404	0.3609	0.0815	0.0088
RegBasic	.441	.475	.496	.398	0.3815	0.0969	0.0155
RegSum	.466	.503	.518	.479	0.3875	0.0983	0.0150
Greedy-KL	NA				0.3823	0.0898	0.0126
Peer 65	NA				0.3762	0.0896	0.0151
Submod	NA				0.3918	0.0935	0.0139
DPP	NA				0.3979	0.0962	0.0157

Table 4.1: Performance comparison of different methods on summarization, evaluated on the DUC 2004 dataset. We also show the performance of summary keyword identification, which reports F_1-score by comparing the top k words to the gold standard keywords that appear in at least i human summaries (G_i).

Both RegBasic and RegSum outperform Prob, LLR and MRW significantly on R-1, R-2 and R-4 by Wilcoxon signed-rank test (paired). The advantage of RegBasic confirms the advantage of including location related information for summarization. RegSum gives 0.006 improvement on R-1 compared to RegBasic; while RegSum and RegBasic perform similarly in terms of R-2 and R-4.

In general, better performance in keyword identification has positive correlations with better performance in summarization: the Spearman correlation between 100 vs G_1 and R-2 is 0.97. However, there are two observations that merit attention.

First, although RegBasic performs similar to Prob in keyword identification (also see Figure 3.6), it improves the performance in summarization. This might be due to two reasons. First, the inclusion of position features boosts the weights of the words that appear close to the beginning of the documents, which is helpful to identify more informative sentences. Second, the learned weights provide a better way of estimating word importance compared to word probability, where the appropriately scaled learned weight can be interpreted as the probability of a word be used in human summaries. Indeed, better word weight estimation involves two aspects: better word ranking (i.e., keyword identification) and assign more proper weights. Thus, a plausible explanation on this phenomenon is: Prob and RegBasic have similar performance in ranking words, while RegBasic assigns better weights and triggers the selection of more informative sentences. RegSum has advantages on all three aspects.

Second, even though RegSum performs better than RegBasic for keyword identification, the advantage of that over RegBasic in summarization is limited. One possible explanation might be due to the setting of extractive summarization, where we have to select content on the sentence level. This is less flexible compared to abstractive summarization settings, where systems have more freedom in deciding the words or phrases to be included. Indeed, the task of summary keyword identification can be regarded as a very extreme case of abstractive summarization, where the summaries are formed using keywords only. In this scenario, better keyword identification is very likely to lead to better "summaries".

So, "Does better identification of summary keywords indicate better summarization results?" The answer is "most of the times, but not always." "Does better estimation of word importance help to generate better summaries?" The answer is "more likely to be true than the previous answer." Indeed, by conducting better estimation of word importance, we can rank the words better (i.e., better summary keyword identification) and assign more appropriate weights to words.

We also compare RegSum and RegBasic with two competitive baseline systems (Peer 65 (Conroy et al., 2004), Greedy-KL[3] (Section 3.5.3)) and two state-of-the-art

[3] The performance of the Greedy-KL system is slightly different from what we reported in the original paper (Hong and Nenkova, 2014a). Originally, we incorrectly normalized the word probability distribution for each sentence and then take the sum of sentence level probability to compute the summary level one. The new result is 0.45% higher on ROUGE-2 than the original one. We also use summaries from the new Greedy-KL system in our analysis in Section 4.2.

systems (Submod (Lin and Bilmes, 2012), DPP (Kulesza and Taskar, 2012)).[4] A detail description of Peer 65, Submod and DPP can be found in Section 4.2.2.

On R-2, RegSum performs significantly better than Peer 65 and Greedy-KL. RegBasic does not outperform these two systems significantly. There is no significant differences between the two state-of-the-art systems and RegBasic/RegSum on R-2 and R-4. DPP performs significantly better than RegSum and RegBasic on R-1, but this model optimizes on unigram F_1-score while training. In summary, by employing our estimates of word importance, GreedySum is able to generate summaries with a performance comparable to the state-of-the-art.

Real-valued weighting vs Binary weighting

Our previous methods all assign real values to words in the input. It has been shown in prior work that for LLR weighting (Section 3.4.2), assigning weight 1 to a small set of keywords and assigning weight 0 to all other words outperforms using the LLR weights themselves (Gupta et al., 2007). In this subsection, we empirically test whether this idea can be successfully generalized to other word weighting functions.

Formally, let h_M denote the word weighting function of method M and let k be the number of keywords we want to select. Let $rank(w)$ denote the rank of word w sorted by $h_M(w)$ in descending order. The *binary weighting* function $b_M(w,k)$ for method M is defined as:

$$b_M(w,k) = \begin{cases} 1 & rank(w) \leq k \\ 0 & \text{otherwise} \end{cases}$$

By doing binary weighting, we assign weight 1 to the keywords and weight 0 to other words. Hence, for this method, what matters are not word weights, but the keywords that have been identified.

[4] The results are slightly different from the ones reported in Lin and Bilmes (2012) and Kulesza and Taskar (2012), because we truncated to 100 words, while they truncated to 665 bytes.

We compare $h_M(w)$ and $b_M(w, k)$ of five methods: word probability (Prob), log-likelihood ratio (LLR), Markov random walk (MRW), RegBasic and RegSum. We use GreedySum to generate summaries, where $h_M(w)$ and $b_M(w, k)$ are used as the weighting function $h(w)$ in Step 1. The experiments are conducted on the DUC 2004 dataset.

Figure 4.1 compares the R-2 of the summaries. For $b_M(w, k)$, we plot the change of performance with different number of keywords selected. For $h_M(w)$, we use flat lines to represent the performance.

For all three unsupervised methods (Prob, LLR, MRW), selecting the top k keywords and employing binary weights almost always outperforms using real-value weights when $15 \leq k \leq 150$. This suggests that, even though how to select k remains an open problem and relies on the property of the input, select a k that is neither too big nor small is a good idea. Further, it suggests that for summarization systems where the weights are estimated by simple heuristics, it might be better to select keywords first and assign the same values to those keywords.

Below we explain why binary weighting may perform better for unsupervised methods. By employing the original methods, the weights assigned to the top few words are often too high. For instance, the weight of "Ranariddh" is 2.3 times as high as that of "Sihanouk" when ranked by word frequency (see Table 4.2). However, they are both among the most important words of the input, which appear in all human summaries (see Table 4.3). This problem is more severe for LLR weighting. Binary weighting helps to remedy this disadvantage, though not perfectly.

On the other hand, assigning binary weights does not help for RegSum, the system where the weights are learned by a variety of features. Among all k we have examined, none of them achieves an improvement over using the real-valued weights in terms of ROUGE-2 (see Figure 4.1). Indeed, the learned weight can be regarded as the probability that one word appears in human summaries (see Table 4.2). The disadvantage of unsupervised methods does not exist any more. In this way, binary

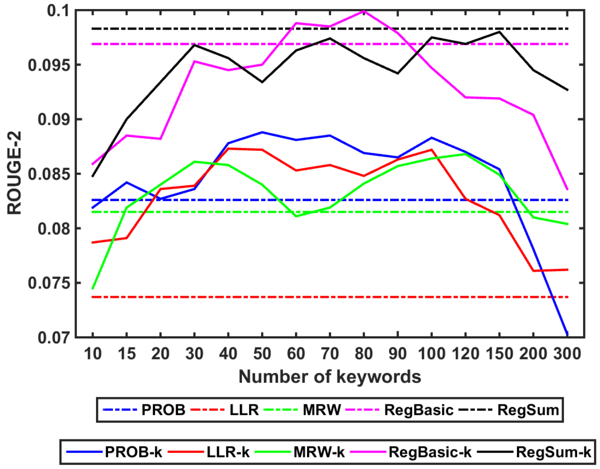


Figure 4.1: Performance of different systems by real-valued weighting (dashed lines) and binary weighting (solid line) on the DUC 2004 dataset. x-axis shows the number of keywords selected.

Prob		LLR		MRW		RegBasic		RegSum	
word	w	word	w	word	w	word	w	word	w
Hun	93	Hun	539.7	Sen	0.0215	Hun	0.990	Hun	0.984
Sen	88	Sen	517.2	party	0.0170	Cambodia	0.983	opposition	0.973
party	52	Ranariddh	277.5	government	0.0133	opposition	0.975	party	0.972
Ranariddh	46	Rainsy	195.1	Ranariddh	0.0114	Rainsy	0.972	Ranariddh	0.965
opposition	41	opposition	122.8	Rainsy	0.0107	assembly	0.971	Cambodia	0.965
government	41	Sihanouk	117.1	Hun	0.0089	Sam	0.961	Sen	0.951
Rainsy	33	Cambodia	113.7	form	0.0073	Ranariddh	0.959	Sam	0.945
Sam	32	Sam	113.4	opposition	0.0073	Sihanouk	0.954	arrest	0.941
Cambodia	28	Norodom	105.1	assembly	0.0073	government	0.950	Rainsy	0.932
assembly	22	Cambodian	104.5	parties	0.0059	parties	0.926	election	0.931
Cambodian	22	CPP	102.5	return	0.0058	form	0.923	Cambodian	0.922
Sihanouk	20	party	100.9	make	0.0054	Sen	0.919	parties	0.921
parliament	19	Funcinpec	84.4	agreed	0.0050	Norodom	0.915	government	0.919
king	19	assembly	60.2	members	0.0050	strongman	0.913	assembly	0.914
prince	18	prince	59.7	Sihanouk	0.0045	parliament	0.912	Norodom	0.910

Table 4.2: Top 15 words and their weights (w) estimated by Prob, LLR, MRW, RegBasic and RegSum for the input set d30003t of the DUC 2004 data.

Input d30001t

Human Summary A
Prospects were dim for resolution of the political crisis in Cambodia in October 1998. Prime Minister Hun Sen insisted that talks take place in Cambodia while opposition leaders **Ranariddh** and Sam Rainsy, fearing arrest at home, wanted them abroad. King **Sihanouk** declined to chair talks in either place. A U.S. House resolution criticized Hun Sen's regime while the opposition tried to cut off his access to loans. But in November the King announced a coalition government with Hun Sen heading the executive and **Ranariddh** leading the parliament. Left out, Sam Rainsy sought the King's assurance of Hun Sen's promise of safety and freedom for all politicians.

Human Summary B
Cambodian prime minister Hun Sen rejects demands of 2 opposition parties for talks in Beijing after failing to win a 2/3 majority in recent elections. **Sihanouk** refuses to host talks in Beijing. Opposition parties ask the Asian Development Bank to stop loans to Hun Sen's government. CCP defends Hun Sen to the US Senate. FUNCINPEC refuses to share the presidency. Hun Sen and **Ranariddh** eventually form a coalition at summit convened by **Sihanouk**. Hun Sen remains prime minister, **Ranariddh** is president of the national assembly, and a new senate will be formed. Opposition leader Rainsy left out. He seeks strong assurance of safety should he return to Cambodia.

Human Summary C
Cambodia King Norodom **Sihanouk** praised formation of a coalition of the Countries top two political parties, leaving strongman Hun Sen as Prime Minister and opposition leader Prince Norodom **Ranariddh** president of the National Assembly. The announcement comes after months of bitter argument following the failure of any party to attain the required quota to form a government. Opposition leader Sam Rainey was seeking assurances that he and his party members would not be arrested if they return to Cambodia. Rainey had been accused by Hun Sen of being behind an assassination attempt against him during massive street demonstrations in September.

Human Summary D
Cambodian elections, fraudulent according to opposition parties, gave the CPP of Hun Sen a scant majority but not enough to form its own government. Opposition leaders fearing arrest, or worse, fled and asked for talks outside the country. Han Sen refused. The UN found evidence of rights violations by Hun Sen prompting the US House to call for an investigation. The three-month governmental deadlock ended with Han Sen and his chief rival, Prince Norodom **Ranariddh** sharing power. Han Sen guaranteed safe return to Cambodia for all opponents but his strongest critic, Sam Rainsy, remained wary. Chief of State King Norodom **Sihanouk** praised the agreement.

Table 4.3: Human summaries of the input set d30001t on the DUC 2004 dataset. The words "Ranariddh" and "Sihanouk" and shown in **bold**.

weighting only causes a loss of information. We also examine the effectiveness of binary weighting for RegBasic. When $k = 60, 70, 80, 90$, it outperforms the performance of real-valued weighting. However, since the advantage is relatively limited and only holds for a small range of k, it is hard to claim which of these two is better for RegBasic; probably still real-valued weighting.

4.2 Output Comparison

4.2.1 Introduction and Motivation

This section focuses on comparing the output of *generic summarization* systems. One might think this is easy, since all systems report their results in papers, and almost all of them use ROUGE (Lin, 2004) for automatic evaluation. However, it is not as easy as expected, because:

- Different systems are evaluated on different datasets.
- Even if they are evaluated on the same datasets, they are evaluated on different metrics (i.e., different ROUGE arguments).
- Even if they use the same datasets and the same ROUGE arguments, the performance of the previous system is represented as a single number (c_0) which is averaged over the test set. Therefore, the advantage of one system over the other is often shown by comparing the performance (or confidence interval) of the current system and c_0 of the previous system. This is not as reliable as comparing systems using a paired statistical significance test.

In this section, we aim to remove these stumbling blocks in evaluation of generic summarization. We present a repository[5] of summaries produced by six state-of-the-art summarization systems and six popular baseline systems. The repository was

[5] This repository is publicly available: `https://www.seas.upenn.edu/nlp/corpora/~sumrepo.html`

made available in 2014. The systems we include can all be (or broadly) regarded as extractive summarization systems, which directly selects sentences from the original input. The six state-of-the-art systems were developed between 2009 and 2014, and have all demonstrated their superiority in the original papers. The input in our repository comes from the DUC 2004 workshop, the latest year in which generic summarization was addressed in a shared task. This dataset is the most popular one for evaluation in generic summarization (Takamura and Okumura, 2009; Lin and Bilmes, 2011; Kulesza and Taskar, 2012; Hong and Nenkova, 2014a). Apart from DUC 2004, systems were also evaluated on earlier years of the DUC data or other data. Notably, many generic summarization systems evaluate their performance on benchmarks designed for query-focused or guided summarization, where these systems simply ignore the specifications (Berg-Kirkpatrick et al., 2011; Woodsend and Lapata, 2012; Almeida and Martins, 2013). However, it is not advisable to do this, since the human summaries used for comparison are designed to answer the query or address the guidance.

Even summarizers evaluated on the same dataset cannot be directly compared in most cases. The reason for this confusion is the number of parameters that one can set for ROUGE. ROUGE evaluates summaries by comparing the n-gram overlap between a summary and a set of gold-standard human summaries. Obviously, the choice of n-gram size will change the score. Furthermore, as ROUGE generates recall, precision and F-measure, different researchers have reported a different combination of the above, despite that recall is the most reasonable choice. To make matters even more disorienting, the overlap with the gold-standard can be computed with function words preserved or excluded and words stemmed or unstemmed; the system summary can be left as is, truncated to specific words or specific bytes while evaluation.

System Peer 65, the best system in the DUC 2004 official evaluation, is a specific case in point. A quick survey of the literature shows that its ROUGE-1 (unigram recall) has been reported as 0.391 (Manna et al., 2012), 0.383 (Lin and Bilmes, 2011), and 0.308 (Conroy et al., 2006a), depending on the parameters used for evaluation.

While testing, we adopt the recommendations from recent findings on summarization evaluation (Owczarzak et al., 2012; Rankel et al., 2013) to fix the ROUGE parameters to those that lead to the highest agreement with manual evaluation. Specifically, we report ROUGE-1, -2, -4, which have been shown to have good and complementary behaviour (see Section 2.2.2 for details). Furthermore, we perform paired tests for statistical significance to establish the superiority of one system over another (Rankel et al., 2011), and find that the best systems developed after CLASSY 04 are not significantly different from each other in terms of bigram recall (ROUGE-2). In summary, our work successfully tackles the three difficulties we have mentioned above.

We further analyze why the top systems achieve a similar performance. It is conceivable that despite using markedly different approaches, the systems may end up choosing the same words, same phrases or even the same sentences. To quantify the extent to which this overlap in selection choices occurs, we compare the summaries produced by the best systems at three levels of granularity: words, summary content units (SCU) (Nenkova et al., 2007) and sentences. Experiments show that the overlap in content across summaries is lower than expected. Our findings suggest that there is high potential of doing summary combination, which will be discussed in Chapter 6.

We first describe the baseline and the state-of-the-art systems whose output are included in our repository, then compare the differences between these systems (Section 4.2.2). In Section 4.2.3, we evaluate the performance of these systems and compute the statistical significance between systems. Finally, we investigate the diversity between summaries from our collection of the state-of-the-art systems (Section 4.2.4).

4.2.2 Systems

Baseline Systems

Below we describe the baseline systems. The first four are adapted from four very popular summarization methods, which achieved impressive performance at the time they were published. These methods are also often used as baselines nowadays. Here we implement these methods using the GreedySum summarizer (Section 4.1.2) which iteratively selects the most importance sentence. The cut-off of cosine similarity that helps to exclude redundant sentences is set to 0.5.[6] The differences between those approaches is thus how to estimate sentence importance. The fifth summarizer (Greedy-KL) is a much stronger baseline compared to the first four, but not as often used. The sixth baseline system is Peer 65 (CLASSY 04), the best system in the official DUC 2004 evaluation.

ProbSum: A simple yet powerful approach for multi-document summarization is to approximate the importance of words with their probability in the input, then select sentences with high average word probability (Nenkova and Vanderwende, 2005). Here we adopt this idea, where word probability is the $h(w)$ in GreedySum.

LLRSum: Another powerful method of weighting words is the application of the log-likelihood ratio (LLR) test (see Section 3.4.2 for a description of the LLR test). Since LLR follows a χ^2 distribution, it provides a natural way to select keywords (a.k.a, *topic signatures*, *topic words*) by setting a predefined confidence level cutoff. Summarizer based on this achieved the state-of-the-art performance on the DUC 2004 and DUC 2006 data when it was designed (Conroy et al., 2006a). Here we consider words to be topic words if their χ^2 statistic

[6] This is the original setting in our repository (Hong et al., 2014), which is different from Section 4.1 (0.4). Since the difference in performance is very small (smaller than 0.001 for ProbSum and RegSum on R-2), we keep the original setting and use the original summaries in the repository.

exceeds 10, which corresponds to a 99.84% confidence level. This setting is the same as Gupta et al. (2007) and similar to the value of 10.0 in Lin and Hovy (2000). While being used in GreedySum, $h(w)$ is equal to 1 if w is a topic word, otherwise it is equal to 0.

Centroid: "*A centroid is a set of words that are statistically important to a cluster of documents*" (Radev et al., 2004b), where oftentimes all words in the document set are used (Radev et al., 2004a). The centroid score of one sentence in the input is equal to the cosine similarity between the sentence and the centroid vector (Radev et al., 2000), where the vectors are represented using bag-of-words models with words weighted by TF*IDF. We compute these scores using the MEAD system (Radev et al., 2004a),[7] then use GreedySum to produce summaries (starting from step 3). $Score(s_i)$ takes the centroid score.

Cont. LexRank: LexRank (Erkan and Radev, 2004) is one of the most popular graph-based methods for summarization. The input text is represented as a graph $G(V, E)$, where V is the set of sentences in the input. In the original LexRank algorithm, there is an edge e_{ij} between two nodes v_i and v_j iff the cosine similarity between them is above a certain threshold. Here we employ continuous LexRank (also introduced in Erkan and Radev (2004)), thus we do not need to tune the threshold manually. The edge weight e_{ij} is equal to the cosine similarity between the sentence vector of v_i and v_j. Sentence importance is calculated by running the PageRank (Page et al., 1998) algorithm on the graph (see Section 3.4.3 for an introduction of the algorithm). The LexRank score of each sentence is derived from MEAD directly. We then use GreedySum to produce summaries (starting from step 3), where $Score(s_i)$ takes the LexRank score.

[7] Note that the MEAD system by default scores sentence importance as a linear combination of the centroid score and the position score of the sentence in the document. We do not use position information here.

Greedy-KL: This system is described in Section 3.5.3.

CLASSY 04 (Peer 65): This was the best system in the official DUC 2004 evaluation (Conroy et al., 2004). It is often used as a comparison system by developers of novel summarization methods. It employs a Hidden Markov Model (HMM), using topic signatures as the only feature. Since HMM was used, the probability of one sentence being selected in the summary also depends on the importance assigned to its adjacent sentences in the input document. The HMM model was trained on the DUC 2003 dataset. A subset of non-redundant sentences with highest scores is selected using non-negative matrix factorization algorithm. It is worth noting that there is a linguistic preprocessing (sentence trimming) component in this system.

State-of-the-art Systems

The state-of-the-art systems are described in alphabetical order. Apart from RegSum, the output of the other systems are contributed by their developers.

CLASSY 11: This (Conroy et al., 2011) is the successor of CLASSY 04, developed to handle query-focused summarization. It performed the best according to *overall responsiveness* (i.e., a metric that evaluates both content quality and linguistic quality) in the official TAC 2011 evaluation of guided summarization. The original system first scores sentences based on bigrams, where topic signatures (Section 3.4.2) and overlap with query terms are used as features to estimate bigram weights. The sentence selection algorithm is the same as CLASSY 04. Two major changes to CLASSY 11 were made for this study. First, for ease of comparison with CLASSY 04, the linguistic preprocessing component in CLASSY 04 was used. Second, the sentences are scored based on unigrams, weighted by latent semantic analysis (LSA) (Deerwester et al., 1990). Davis et al. (2012) describe how are the LSA weights computed.

DPP: Determinantal point processes (DPPs) (Kulesza and Taskar, 2012) are probabilistic models of sets which balance the selection of important information and diverse groups of sentences within a given length. Specifically, DPPs combine a per-sentence quality model that prefers relevant sentences with a global diversity model encouraging non-overlapping content. This setup has several advantages. First, by treating these opposing objectives probabilistically, there is a rigorous framework for trading off between them. Second, the sentence quality model can depend on arbitrary features, and its parameters can be efficiently learned from reference summaries via maximum likelihood training; in contrast, most standard summarization techniques are tuned by hand. Finally, because a DPP is a probabilistic model, at test time it is possible to sample multiple summaries and apply minimum Bayes risk decoding, thus improving ROUGE scores. The DPP model in this work is trained on the DUC 2003 data to optimize the unigram F-score.

ICSISumm: The ICSISumm system (Gillick et al., 2008; Gillick et al., 2009) adopts an Integer Linear Programming (ILP) framework, finding the globally optimal summary rather than greedily choosing sentences according to their importance. Even though ILP is NP-hard, the exact solutions can be found by a standard ILP solver in a fairly fast fashion. Specifically, this system optimizes the coverage of key bigrams (i.e., bigrams that appear in at least three input documents), which are weighted by their document frequency in the document collection. It also includes a linguistic preprocessing component, which removes the irrelevant words or phrases from the input sentences.

OCCAMS: This system (Davis et al., 2012; Conroy et al., 2013) is a successor of CLASSY 11. It employs LSA to compute word weights, then employs a sentence selection algorithm based on two combinatorial problems: the budgeted maximal coverage (BMC) problem and the knapsack problem. The authors first regard summarization as a BMC problem, which optimizes the coverage of word weights corresponded to a length constraint. Let L denote the desired summary length, OCCAMS first selects a set of candidate sentences of total length $5L$ by applying a greedy algorithm to solve the BMC problem (Khuller et al., 1999). The authors then formulate summarization as a knapsack problem, which maximizes the total cost covered by sentences, given the length constraint L. A dynamic programming based Fully Polynomial Approximation Scheme (FPTAS) is used to solve the problem and produce the final summary. As was done with CLASSY 11, OCCAMS also uses the CLASSY 04 linguistic preprocessing component for this study, which is different from the original preprocessing component used in David et al. (2012) and Conroy et al. (2013).

RegSum: The RegSum system (see Chapter 3) employs a supervised model to improve the estimation of word importance. Similar to ProbSum and LLRSum, we use GreedySum to generate summaries, only changing the way of weighting words. Here we use the summaries from the original paper (Hong and Nenkova, 2014a), where the cosine similarity cut-off of checking redundancy is set to 0.5.

Submod: Treating multi-document summarization as a submodular maximization problem has proven successful (Lin and Bilmes, 2011) and has spurred a great deal of interest in this line of research (Sipos et al., 2012; Morita et al., 2013; Dasgupta et al., 2013). The advantage of using a submodular function to estimate summary importance lies in the fact that there exists an efficient algorithm for incrementally computing the importance of a summary with a performance guarantee on how close the approximate solution will be to the globally optimal one.

We collect summaries from Lin and Bilmes (2012), where they employ structure learning to produce the submodular functions. They first learn a mixture

	Unsupervised	Supervised
Greedy	ProbSum (word) LLRSum (word) Centroid (sentence) Cont. LexRank (sentence) Greedy-KL (sentence) CLASSY 11 (word)	CLASSY 04 (word) RegSum (word)
Approximately Global Inference	OCCAMS (word)	Submod (sentence)
Global Inference	ICSISumm (bigram)	DPP (summary)

Table 4.4: Comparison between systems. The smallest units that these systems use to estimate content importance are shown in brackets.

of submodular "shells" in a max-margin structured prediction setting. Then a mixtures of the shells can be instantiated to generate a more complex submodular function.

Differences Between Systems

By reviewing the baseline and more recent systems, we have observed a shift from unsupervised to supervised methods. We have also observed a shift from greedy based methods that iteratively selects the most importance sentence, to global inference based methods that directly selects a set of sentences. There are also systems which employ greedy based methods which approximately optimize a function that describes the overall summary quality (approximately global inference). Moreover, the systems we include also differ in the smallest unit that they use to assign importance scores. Some systems start from weighting word importance, some start from estimating sentence importance. DPP directly estimates the summary importance by sampling multiple summaries and apply minimum Bayes decoding. Table 4.4 compares the summarization systems in our repository.

Other Summarization Methods

In addition to the systems we have included, many methods have been developed for extractive summarization. Among them, approaches that utilize topic models (Blei et al., 2003) are popular in recent years (Daumé III and Marcu, 2006; Haghighi and Vanderwende, 2009; Celikyilmaz and Hakkani-Tur, 2010; Celikyilmaz and Hakkani-Tür, 2011). These methods represent the input into different semantic topics, where each topic is represented by words and their corresponding weights. Compared to the methods in our repository, methods based on topic models take advantage of the hidden semantic structure of the input documents. Moreover, these methods often have an explicit topic model representation of the single documents in the input, which is helpful to exploit the similarity or diversity between the documents of the same input.

A large variety of methods have been developed for extractive summarization and many survey papers have reviewed these methods (Das and Martins, 2007; Gupta and Lehal, 2010; Nenkova and McKeown, 2012). In contrast, less work has focused on abstractive summarization, where the summaries are generated freely (i.e., not necessarily use the input sentences verbatim). For abstractive based systems, sentence compression is the most widely used technique. Compression based systems remove irrelevant details from input sentences, which makes it feasible to include more content into the summaries. Two general strategies are often used for compressive summarization. The first strategy conducts sentence compression and extraction one after another (Wang et al., 2013; Li et al., 2013a), the second performs sentence compression and extraction simultaneously (Berg-Kirkpatrick et al., 2011; Woodsend and Lapata, 2012). For compressive based systems, sentence compression is by no means less important than extraction. Incorrect sentence compression not only affects linguistic quality, but also affects content quality because the compressed sentences may not be understood by humans. Different from sentence compression as a stand-alone task, the compression model in summarization systems should decide whether or not a word should be retained based on its linguistic quality as well as its importance for summarization.

Other techniques used in abstractive based systems include sentence fusion, sentence revision, sentence simplification, etc. A review of these techniques can be found in Chapter 4 in Nenkova and Mckeown (2011).

4.2.3 ROUGE Score and Significance Test

Performance Comparison

We use three ROUGE metrics for evaluation: ROUGE-1, -2, -4 (R-1, R-2, R-4), all with stemming and stopwords included.[8] These three metrics have different advantages, as described in Section 2.2.2. We use ROUGE-2 as the main evaluation metric. Each summary is truncated to 100 words automatically by ROUGE.[9]

Table 4.5 shows the performance of all systems, sorted by R-2 in ascending order. Greedy-KL performs much better than ProbSum, LLRSum, Centroid and Cont. LexRank on all three ROUGE scores. It even achieves R-1 higher than CLASSY 04 and CLASSY 11. It is also marginally better than CLASSY 04 in terms of R-2. The system with the highest R-1 is DPP, which exceeds that of CLASSY 04 by 0.0237. It also achieves a better performance on R-1 than all other systems by at least 0.01, except for Submod where the difference is 0.0059. Note that this system's edge over the other approaches is not so clear-cut on R-2 and R-4, which likely is a reflection of the fact that DPP optimizes unigram F-measure during the learning process. ICSISumm optimizes the coverage of bigram document frequency, and achieves the highest performance on R-2 and R-4. The state-of-the-art systems perform extremely

[8] ROUGE-1.5.5 with the parameters: -n 4 -m -a -l 100 -x -c 95 -r 1000 -f A -p 0.5 -t 0 (-n 4 -m -l 100 -x -a gives the same result).

[9] In the official DUC 2004 evaluation, summaries were truncated to 665 bytes. This meant that systems would be truncated to different numbers of words for evaluation. This is disturbing, since the variation in length has a noticeable impact on automatic evaluation results. Therefore, we truncate the summaries to 100 words while evaluation.

System	ROUGE-1	ROUGE-2	ROUGE-4
Cont. LexRank	0.3595	0.0747	0.0082
Centroid	0.3641	0.0797	0.0121
ProbSum	0.3530	0.0811	0.0100
LLRSum	0.3588	0.0815	0.0103
CLASSY 04	0.3762	0.0896	0.0151
Greedy-KL	0.3823	0.0898	0.0126
CLASSY 11	0.3722	0.0920	0.0148
Submod	0.3918	0.0935	0.0139
DPP	0.3979	0.0962	0.0157
RegSum	0.3857	0.0975	0.0160
OCCAMS	0.3850	0.0976	0.0133
ICSISumm	0.3841	0.0978	0.0173

Table 4.5: Performance comparison between systems. We report ROUGE-1, ROUGE-2 and ROUGE-4 of systems on the DUC 2004 dataset.

close on R-2, which only differing within a range of 0.0058. CLASSY 04 has a strong performance on R-4 (0.0151), with no system exhibits a significantly better performance on this evaluation.

Significance Test

So far we have discussed the ROUGE score of different systems. However, it would be too hasty to claim one system's advantage over the other simply based on this, unless the difference between systems is huge. Indeed, the fact that system A has 0.003 higher ROUGE-2 than system B does not necessarily mean A is a better system. It is also possible that this happens by chance.

Rather than simply comparing the average scores, another way of comparing systems is adopted by many researchers. When ROUGE is reporting performance (denoted as p), it also provides a 95% confidence interval ($p_1 \leq p \leq p_2$), derived by bootstrap resampling (Davison, 1997). Researchers would thus compare the performance of a previous system (denoted as p_0) with the confidence interval of the newer system. If $p_0 < p_1$, then they would claim that their system is better.

Even though this method is better than comparing p and p_0 directly, it still has problems. Indeed, the final ROUGE score is derived by taking the average ROUGE scores over different input sets. These input sets have completely different difficulty for automatic systems (Nenkova and Louis, 2008), and should not be mixed up while comparison. Thus, a better idea is to pair up summaries over the same input, which can be accomplished by paired test for statistical significance. Rankel et al. (2011) empirically verified that paired test is a good idea. There, they demonstrated the advantage of conducting paired test over unpaired test, where the later one does not consider the differences between inputs. Their experiments are based on the assertion that human summarizers would always outperform machine summarizers. They show that by using paired test, it is easier to claim that a human summarizer is significantly better than a machine summarizer.

Note that paired test can be performed only if we have the output from both systems. Our repository not only makes it possible for us to do this, but also makes it easier for future researchers to conduct paired comparison.

Hence, we conduct *paired* two-sided Wilcoxon signed-rank (WSR) tests between each pair of the systems, as advocated in Rankel et al. (2011). The results between each pair of the top eight systems are presented in Table 4.6. The eight systems include our six state-of-the-art systems and our two best baselines (Greedy-KL, CLASSY 04). The order of systems in these tables are listed according to the order they are described in Section 4.2.2. A plus sign before the p-value indicates that the system in the row performs better than the one in the column, a minus sign indicates the opposite. The other four baseline systems are not shown in the tables; all of them perform significantly worse than the six state-of-the-art systems on R-2.

ROUGE-1	CLASSY 04	CLASSY 11	DPP	ICSISumm	OCCAMS	RegSum	Submod
Greedy-KL	(+) 0.292	(+) 0.133	(-) **0.013**	(-) 0.528	(-) 0.809	(-) 0.398	(-) 0.063
CLASSY 04		(+) 0.708	(-) **0.001**	(-) 0.114	(-) 0.130	(-) 0.080	(-) **0.008**
CLASSY 11			(-) **1e-4**	(-) **0.032**	(-) **0.006**	(-) **0.029**	(-) **0.001**
DPP				(+) **0.011**	(+) **0.005**	(+) **0.016**	(+) 0.072
ICSISumm					(-) 0.971	(-) 0.791	(-) 0.135
OCCAMS						(-) 0.778	(-) 0.193
RegSum							(-) 0.216

(a) Statistical significance between the performance of systems on ROUGE-1.

ROUGE-2	CLASSY 04	CLASSY 11	DPP	ICSISumm	OCCAMS	RegSum	Submod
Greedy-KL	(-) 0.988	(-) 0.518	(-) 0.060	(-) **0.007**	(-) **0.039**	(-) **0.039**	(-) 0.549
CLASSY 04		(-) 0.521	(-) 0.133	(-) 0.071	(-) 0.057	(-) **0.048**	(-) 0.317
CLASSY 11			(-) 0.164	(-) 0.055	(-) 0.130	(-) 0.116	(-) 0.655
DPP				(-) 0.482	(-) 0.652	(-) 0.579	(+) 0.317
ICSISumm					(+) 0.791	(+) 0.758	(+) 0.186
OCCAMS						(-) 0.976	(+) 0.156
RegSum							(+) 0.126

(b) Statistical significance between the performance of systems on ROUGE-2.

ROUGE-4	CLASSY 04	CLASSY 11	DPP	ICSISumm	OCCAMS	RegSum	Submod
Greedy-KL	(-) 0.359	(-) 0.238	(-) **0.032**	(-) **0.002**	(-) 0.514	(-) **0.026**	(-) 0.955
CLASSY 04		(+) 0.696	(-) 0.623	(-) 0.180	(+) 0.754	(-) 0.441	(+) 0.739
CLASSY 11			(-) 0.469	(-) 0.081	(+) 0.267	(-) 0.440	(+) 0.518
DPP				(-) 0.229	(+) 0.122	(-) 0.906	(+) **0.040**
ICSISumm					(+) **0.005**	(+) 0.356	(+) **0.009**
OCCAMS						(-) 0.121	(-) 0.421
RegSum							(+) 0.066

(c) Statistical significance between the performance of systems on ROUGE-4.

Table 4.6: p-values from paired two-sided Wilcoxon signed-rank test on ROUGE-1, ROUGE-2, ROUGE-4. $p < 0.05$ is shown in **bold**. A plus (minus) sign before the p-value indicates that the system in the row (column) performs better than the one in the column (row).

On ROUGE-1, only DPP and Submod show significant improvement over CLASSY 04. DPP performs significantly better than all but one system (Submod) on R-1.

On ROUGE-2, there are no significant differences between the best six systems. RegSum is the only one that shows a significant improvement over CLASSY 04 ($p = 0.048$). The advantage of OCCAMS ($p = 0.057$) and ICSISumm ($p = 0.071$) over CLASSY 04 is close to significant. Three state-of-the-art systems (ICSISumm, OCCAMS, RegSum) perform significantly better than Greedy-KL.

On ROUGE-4, no recently developed systems outperforms CLASSY 04 significantly. ICSISumm has the best R-4, but the p-value for this over CLASSY 04 is only 0.180. DPP, ICSISumm and Regsum are significantly better than Greedy-KL.

In summary, we have so far established that the state-of-the-art systems developed in recent years comfortably outperform several standard baselines. They also work better than the best system in the DUC 2004 workshop (CLASSY 04). We demonstrate that the Greedy-KL baseline performs very well, at times on par with the best systems existing to date. Given its simplicity and excellent performance, Greedy-KL should be used as a baseline in future studies. Despite using markedly different summarization methods, the state-of-the-art systems that we compare perform similarly in automatic evaluation results. We next investigate why this happens.

4.2.4 Overlap Between Summaries

In this section, we examine if the similar performance between systems is because they select the same content. We study the overlap between the produced summaries at the sentence level and the word level. For a subset of inputs, we also study the overlap of manually annotated summary content units (SCUs), which are the basic units in the Pyramid Method (Nenkova et al., 2007). Here we study the overlap between the top eight systems, which include six state-of-the-art systems and the two best baselines.

Overlap at the Sentence Level

To compute the sentence overlap between summaries, we first find the original input sentences that the summary sentences correspond to. This is easy, since the systems in our repository can all be broadly regarded as extractive summarization systems. We then compute the overlap of the input sentences. Our procedure is as follows:

For each sentence in a summary S, we find its corresponding sentence in the input, where an exact match happens most of the times. When an exact match cannot be found, there are two possibilities. First, it is an incomplete last sentence which was truncated by the summarizer in order to meet the 100 word limit. When this happens, we simply ignore the last sentence. Second, it was due to different word tokenization or different preprocessing (e.g., compression). When this happens, we find the most similar sentence in the input, measured by cosine similarity between the binary bag-of-word vectors of the input and the summary sentence. Using the above approach, we form a set S', where S' include the input sentences that corresponds to the summary sentences in S.

We use Jaccard coefficient to quantify the degree of sentence overlap between summaries. Suppose for two systems A and B, we have formed two set of input sentences A_I and B_I, which corresponds to the summary sentences produced by A and B over I. The Jaccard coefficient between A and B over I is:

$$J(A_I, B_I) = \frac{|A_I \bigcap B_I|}{|A_I \bigcup B_I|} \tag{4.3}$$

Let $D = \{I_1, I_2, \ldots, I_n\}$ denote a dataset that include n input sets, the system level Jaccard coefficient between A and B is:

$$J(A, B) = \frac{1}{n} \cdot J(A_{I_i}, B_{I_i}) \tag{4.4}$$

Table 4.7 shows the Jaccard coefficients at the sentence level between each pair of systems. The degree of overlap surprisingly low. The lowest overlap at the sentence-level is between CLASSY 04 and CLASSY 11, with a Jaccard coefficient of only

	CLASSY 04	CLASSY 11	DPP	ICSISumm	OCCAMS	RegSum	Submod
Greedy-KL	0.077	0.130	0.085	0.110	0.078	0.103	0.169
CLASSY 04		0.003	0.075	0.082	0.003	0.136	0.050
CLASSY 11			0.073	0.092	0.169	0.060	0.070
DPP				0.143	0.071	0.124	0.082
ICSISumm					0.083	0.149	0.038
OCCAMS						0.090	0.050
RegSum							0.113

Table 4.7: Sentence overlap between summaries from different systems by Jaccard coefficient.

0.003. OCCAMS and CLASSY 11 extract the most similar sets of sentences, with a Jaccard coefficient of 0.169. The coefficient between two of the best systems—DPP and ICSISumm—is 0.143.

Overlap at the Word Level

We next compute the word overlap between summaries. Here we compute the Jaccard coefficient between the sets of unique words of two summaries for a given input, with stemming and stopwords included. This is consistent with the ROUGE setting that we use. The system level Jaccard coefficient is equal to the average of summary level Jaccard scores among the input sets (see equation 4.4).

Table 4.8 shows the Jaccard coefficients between systems at the word level. The coefficients range between 0.206 (CLASSY 04 vs CLASSY 11) and 0.407 (CLASSY 04 vs RegSum), with the average among all system pairs being 0.318. The value 0.318 can be interpreted as: among all unique words in two summaries, no more than a third appear in both of them. The average coefficient is smaller if the stopwords are excluded (0.282) or stemming is not performed (0.299).

	CLASSY 04	CLASSY 11	DPP	ICSISumm	OCCAMS	RegSum	Submod
Greedy-KL	0.294	0.331	0.332	0.353	0.267	0.359	0.374
CLASSY 04		0.206	0.300	0.315	0.208	0.407	0.289
CLASSY 11			0.311	0.295	0.385	0.262	0.317
DPP				0.387	0.307	0.372	0.367
ICSISumm					0.302	0.375	0.307
OCCAMS						0.274	0.292
RegSum							0.366

Table 4.8: Word overlap between summaries from different systems by Jaccard coefficient. Here we include stopwords and perform stemming.

Overlap at the SCU level

Sentence and word overlap do not directly reflect the overlap in semantic content expressed in the summaries. The low overlap can be due to the reason that two summaries express the same meaning using different sentences, through different wordings. To better investigate the overlap of information between summaries, we compute the overlaps of SCUs using the Pyramid Method (Nenkova et al., 2007). SCUs are defined as semantically motivated sub-sentential units. They are annotated manually and map together all expressions of the same content, even if the wording across summaries differs. First, we manually create the "pyramids" for 10 input sets in DUC 2004; that is, we find the SCUs expressed in the four human written abstracts. Then we identify which SCUs are expressed in the summaries of five of the automatic systems—CLASSY 04, Greedy-KL, DPP, ICSISumm, and RegSum—for those inputs. SCUs conveyed in truncated last sentences are also included in the calculations of the overlap.

On average there are 33.7 SCUs expressed in human summaries. On average each SCU is expressed 1.78 times across the human summaries, with few SCUs repeated frequently and most appearing in only one of them. Table 4.9 shows the average number of SCUs from the reference human summaries that are also expressed in the machine summaries. The number of unique SCUs covered by the five machine summaries is 15.0 on average. There are no significant differences between systems, evaluated by the Pyramid score (Table 4.9).[10]

	CLASSY 04	Greedy-KL	DPP	ICSISumm	RegSum
# SCUs	7.1	7.0	6.9	7.2	7.3
The Pyramid score	0.502	0.485	0.489	0.517	0.521

Table 4.9: The average number of SCUs per summary and their Pyramid scores on the first 10 input sets.

Similar to how we compute the word level and sentence level overlap, we compute the similarity between systems in terms of SCUs using Jaccard coefficient, which is shown in Table 4.10. The similarity score ranges between 0.218 and 0.419, with an average of 0.347. Hence, even in the manual analysis of content overlap, the content covered by two systems are fairly different. Clearly, the systems perform similarly not because they choose the same content—they choose different but equally good content. This finding suggests that it is promising to exploit methods that combines the output from different systems. This idea is also supported by work on fully automatic evaluation for summarization which shows that the combination of different systems' output serves as an excellent reference for estimating summary content quality (Louis and Nenkova, 2013).

	Greedy-KL	DPP	ICSISumm	RegSum
CLASSY 04	0.340	0.265	0.218	0.387
Greedy-KL		0.373	0.341	0.356
DPP			0.390	0.419
ICSISumm				0.384

Table 4.10: The overlap of SCUs by Jaccard coefficient.

[10] Note that the numbers in Table 4.9 and Table 4.10 are slightly different from Hong et al. (2014). This is because we newly compute the Pyramid score for the CLASSY 04 system. For consistency, we recompute the scores for the other four systems.

To more concretely illustrate the differences in summary content, we show the machine summaries for the inputs d30002t and d30003t in Table 4.11 and Table 4.12, respectively. These are the two input sets with the least (0.142) and highest (0.437) degree of overlap in terms of SCUs, measured by Jaccard coefficient. In the two tables, we show the SCUs that are expressed in multiple human summaries. For the input d30002t, 14 unique SCUs are expressed in machine summaries. However, only four of them are expressed in more than one summary.[11] Clearly, there are large differences in the content expressed within those machine summaries. There are 18 unique SCUs expressed for the input d30003t, with 11 of them expressed in at least two summaries. Only three SCUs are expressed in all five machine summaries. In summary, our experiments show that the summaries are of high diversity.

4.3 Conclusion

This chapter focuses on comparing different generic multi-document summarization systems from two perspectives: their methods and their outputs.

To compare different summarization methods, we decouple the summarization process into a word weighting and a summary generation component. We present a modular greedy summarizer, which compares different word weighting methods. Experiment shows that our new word weighting method (described in Chapter 3) outperforms the unsupervised methods and produces summaries comparable to the state-of-the-art. The word weighting method that uses frequency and location related features also performs well, which falls behind the system that uses the full set of features by a small margin. We investigate two strategies of weighing words in summarization: (1) assigning weight to keywords and assigning zero weights to the others, and (2) using original real-valued weights. We show that the former strategy

[11] For the input d30002t, SCUs A, B, C, and D are “Hurricane Mitch brought huge death toll in Central America”, “Slow-moving Mitch battered the Honduran for more than a day”, “Taiwan sent aid to Central American countries”, and “Hurricane caused devastated destruction in Central America”, respectively.

works better if the weights are derived by unsupervised methods, while the latter strategy works better if the weights are learned using our supervised model proposed in Chapter 3. We hypothesize that it is because our model can assign appropriately scaled weights to words.

To compare summaries, we present a repository of summaries produced by a range of systems on the DUC 2004 dataset. This resource allows us to carry out a large scale comparison between recent systems. By doing this, we have tackled the difficulty that different systems were evaluated on different datasets, with different metrics. We have also outlined the methodology for reporting results (i.e., establishing informed choices of ROUGE settings, computing paired statistical significance tests). Furthermore, our repository makes it easy for later work to compare the summaries between two systems directly. Experiment shows that there are no significant differences between the best systems in terms of bigram recall. We also show that diverse contents get selected by the summaries from different state systems. This suggests that summary combination might lead to improvements in content selection, which we will show in Chapter 6.

Input d30002t

CLASSY 04 summary:

Hurricane Mitch paused in its whirl through the western Caribbean on Wednesday to punish [**Honduras with 120-mph (205-kph) winds, topping trees, sweeping away bridges, flooding neighborhoods**]$_D$ and killing at least 32 people. Aid workers struggled Friday to reach survivors of Hurricane Mitch, who are in danger of dying from starvation and disease in the wake of the storm that officials estimate killed more than 10,000 people. Better information from [**Honduras”ravaged countryside**]$_D$ enabled officials to lower the [**confirmed death toll from Hurricane Mitch from 7,000 to about 6,100**]$_A$ on Thursday, but leaders insisted the need for help was growing”. The European Union

Greedy-KL summary:

In Washington on Thursday, President Bill Clinton ordered dlrs 30 million in Defense Department equipment and services and dlrs 36 million in food, fuel and other aid be sent to Honduras, Nicaragua, El Salvador and Guatemala. With the storm seemingly anchored off Honduras, officials in Mexico to the north eased emergency measures on the Caribbean coast of the Yucatan Peninsula, where hundreds of people remained in shelters as a precaution Wednesday night. The Honduran president closed schools and public offices on the coast Monday and ordered all air force planes and helicopters to evacuate people from the Islas de la Bahia, a string of small islands off the country's central coast.

DPP summary:

At 0900 GMT Tuesday, Mitch was 95 miles north of Honduras, near the Swan Islands. Hurricane Mitch cut through the Honduran coast like a ripsaw Thursday, [**its devastating winds whirling for a third day**]$_B$ through resort islands and mainland communities. Mexico reported one death from Mitch last Monday. [**Hurricane Mitch killed an estimated 9,000 people throughout Central America**]$_A$ in a disaster of such proportions that relief agencies have been overwhelmed. He urged the more than 1.5 million Hondurans affected by the storm to help with the recovery effort. [**Taiwan said today it will donate dlrs 2.6 million in relief to Honduras, Nicaragua, El Salvador and Guatemala**]$_C$.

ICSISumm summary:

At least 231 people have been confirmed dead in Honduras from former-hurricane Mitch, bringing the storm's death toll in the region to 357. Pope John Paul II appealed for aid Wednesday for the Central American countries stricken by hurricane Mitch and said he feels close to the thousands who are suffering. [**Officials in Central America estimated Tuesday that about 7,000 people have died in the region**]$_A$. [**That meant the Honduran coast had been under hurricane conditions for more than a day**]$_B$. [**Taiwan said today it will donate dlrs 2.6 million in relief to Honduras, Nicaragua, El Salvador and Guatemala**]$_C$.

RegSum summary:

President Carlos Flores declared [**Hurricane Mitch had set back Honduras development by 50 years**]$_D$. El Salvador reported 239 dead; Guatemala said 194 of its people had been killed. [**Officials in Central America estimated Tuesday that about 7,000 people have died in the region**]$_A$. [**Hurricane Mitch killed an estimated 9,000 people throughout Central America**]$_A$ in a disaster of such proportions that relief agencies have been overwhelmed. Mexico reported one death from Mitch last Monday. The strongest hurricane to hit Honduras in recent memory was Fifi in 1974, which ravaged Honduras Caribbean coast, killing at least 2,000 people.

Table 4.11: Summaries generated by five systems for the input d30002t on the DUC 2004 data. Expressions of the SCUs that appear in multiple summaries are labeled in brackets.

Input d30003t

CLASSY 04 summary:

[**As his lawyers in London tried to quash**]$_K$ a [**Spanish arrest warrant for Gen. Augusto Pinochet**]$_C$, the former Chilean dictator, efforts began in Geneva and Paris to have him extradited. [**Britain has defended**]$_F$ its [**arrest of Gen. Augusto Pinochet**]$_A$, with one lawmaker saying that Chile's claim that the [**former Chilean dictator has diplomatic immunity**]$_E$ is ridiculous. Margaret Thatcher entertained former Children dictator Gen. Augusto Pinochet at her home two weeks before he was arrested in his bed [**in a London**]$_B$ [**hospital**]$_J$, the ex-prime minister's office said Tuesday, amid growing diplomatic and domestic controversy over the move. The Spanish and British governments appeared

Greedy-KL summary:

[**British police**]$_I$, acting on an international [**arrest warrant from Spanish judge**]$_C$ Baltasar Garzon on Friday, detained the ailing general in the [**London clinic**]$_J$ where [**he was recovering from back surgery**]$_H$. [**Pinochet was arrested**]$_A$ [**in London**]$_B$ on Oct. 16 at the instigation of [**Spanish magistrate Baltasar Garzon who is seeking to extradite**]$_G$ the former dictator on [**charges of genocide, terrorism and torture**]$_D$. [**Britain has defended its arrest of Gen. Augusto Pinochet**]$_F$, with one lawmaker saying that Chile's claim that the [**former Chilean dictator has diplomatic immunity**]$_E$ is ridiculous. Despite everything that has been written, this is not a decision for the British government or the Spanish government, British Prime Minister Tony Blair said in an interview published by a Spanish newspaper on Wednesday.

DPP summary:

[**Pinochet, 82, was placed under arrest**]$_A$ [**in London**]$_B$ Friday by [**British police**]$_I$ acting on [**a warrant issued by a Spanish judge**]$_C$. Pinochet, commander-in-chief of the Chilean army until March, [**has immunity from prosecution in Chile**]$_E$ as a senator-for-life under a new constitution that his government crafted. Pinochet was arrested in London on Oct.16 at the instigation of Spanish magistrate Baltasar Garzon who is seeking to extradite the former dictator on [**charges of genocide, terrorism and torture**]$_D$. Europe's top official said Friday that he hoped former Chilean dictator [**Augusto Pinochet would be extradited to Spain**]$_G$ to be tried for crimes committed during his 17-year rule.

ICSISumm summary:

Europe's top official said Friday that he hoped former Chilean dictator [**Augusto Pinochet would be extradited to Spain**]$_G$ to be tried for crimes committed during his 17-year rule. Pinochet was detained in the [**London clinic**]$_J$ [**while recovering from back surgery**]$_H$. [**Pinochet, 82, was placed under arrest**]$_A$ [**in London**]$_B$ Friday by [**British police**]$_I$ acting on [**a warrant issued by a Spanish judge**]$_C$. He was arrested Oct. 16 at the instigation of a Spanish magistrate seeking to extradite him on charges of genocide, terrorism and torture. Spain was the biggest single investor in Chile last year, followed closely by Britain.

RegSum summary:

Pinochet was detained in the [**London clinic**]$_J$ [**while recovering from back surgery**]$_H$. The Spanish and British governments appeared Wednesday to be seeking shelter from the political storm brewing over the possible [**extradition of former Chilean dictator Augusto Pinochet to Spain**]$_G$. [**As his lawyers in London tried to quash**]$_K$ a [**Spanish arrest warrant for Gen. Augusto Pinochet**]$_C$, the former Chilean dictator, efforts began in Geneva and Paris to have him extradited. The 82-year-old Pinochet was arrested Friday at a Spanish magistrate's request. [**Pinochet was arrested**]$_A$ [**in London**]$_B$ on Oct. 16 at the instigation of Spanish magistrate Baltasar Garzon who is seeking to extradite the former dictator [**charges of genocide, terrorism and torture**]$_D$.

Table 4.12: Summaries generated by five systems for the input d30003t on the DUC 2004 data. Expressions of the SCUs that appear in multiple summaries are labeled in brackets.

Chapter 5

Mining Global Knowledge for Summarization

5.1 Introduction

Many[1] traditional summarization systems identify important content solely based on the input (i.e., documents to be summarized). This is true for unsupervised systems (Carbonell and Goldstein, 1998; Erkan and Radev, 2004; Vanderwende et al., 2007) that rely on superficial characteristics of the input such as frequency, location and sentence length. Supervised based systems use documents accompanied by their corresponding human summaries for training, which is different from the input in the test data. However, as the features used by most systems are solely extracted from the documents to be summarized (Yih et al., 2007; Lin and Bilmes, 2010), content importance is still estimated based on input dependent features.

In contrast, human summarizers rely on their prior knowledge in part to identify important content (Mani, 2001a). It has been shown in Endres-Niggemeyer (1998) that professional abstractors search for cue words based on their background knowledge while selecting informative sentences. Prior knowledge also affects the summary

[1] Section 5.3 and Section 5.4 are adapted from Hong and Nenkova (2014a).

produced by non-experts. Recht and Leslie (1988) conducted a study that solicits high school students to produce summaries after reading a half inning of a baseball game. They show that students with a higher degree of familiarity with baseball are able to generate better summaries.

In this chapter, we study whether or not incorporating *global knowledge* is helpful for content selection in summarization. To be specific, we are mainly concerned the *knowledge* that indicates *whether or not a content is likely to be included in summaries, independent of the particulars of the input*. We examine two kinds of global knowledge: (1) dictionary knowledge, and (2) intrinsic word importance.

Dictionaries provide categorical information (e.g., subjectivity, semantic categories) of words. We use dictionaries here, based on the hypothesis that words in certain categories are likely to be globally important or unimportant. In fact, this hypothesis has already been confirmed by our analyses on part-of-speech tags in Section 3.5.2. There, we observe that there are more nouns and past tense verbs in human summaries, while there are fewer present tense verbs and adverbs.

In our work, we make use of two dictionaries: (1) Multi-Perspective Question Answering (MPQA) (Wiebe and Cardie, 2005), which groups words by their polarities and subjectivities; and (2) Linguistic Inquiry and Word Count (LIWC) (Tausczik and Pennebaker, 2007), which groups words by their linguistic and semantic properties. Analysis based on MPQA shows that words with strong subjectivities are unlikely to be used in human summaries (Section 5.3.1). Analysis based on LIWC shows that words which belong to the semantic categories of "death", "anger" and "money" are likely to be used in human summaries (Section 5.3.2).

We then propose a method that directly estimates the intrinsic importance of words to humans. There is reason to believe that doing this is helpful. Consider the following two sentences: "John killed Bill" and "John hit Bill", which appear in the input with the same frequency. If no other information is given, it seems likely that the former sentence will be selected by most abstractors. Our work aims to capture such preference, i.e., assigning higher importance score to “killed” than to “hit”.

To estimate the intrinsic importance of words (a.k.a, *global indicators*), we need a very large corpus. Fortunately, the Linguistic Data Consortium (LDC) released the New York Times (NYT) corpus (Sandhaus, 2008), which includes 650K articles along with their summaries. We estimate the global indicators based on a portion of the corpus. To show the effectiveness of these indicators, we design a summarizer (Blind) that *only* uses the global indicators to weigh words and select sentences. We show that Blind performs comparably to a standard baseline which selects the first L words from the most recent input document. In order to understand *why* some words are globally important/unimportant, we study the MPQA and LIWC categories that these words belong to. This study also investigates what categories of words are likely to be included or avoided in human summaries in general, based on the larger NYT corpus.

These two kinds of global knowledge are used as features in our model that identifies summary keywords (Chapter 3). We have discussed the effectiveness of these features by feature ablation and inclusion experiments (Section 3.6.3). In Section 5.5, we expand our results of the ablation experiment, where we report precision, recall and F_1-score. We also show that removing global knowledge leads to a marginal decrease for summarization. In Chapter 6, we will show that the global indicators are useful in identifying summaries with good content for summary combination. Finally, we show that the global indicators are very helpful in identifying summary keywords that have low frequency in the input; we discuss why this happens.

This chapter is organized as follows. Section 5.2 gives a review of related work. Section 5.3 describes the dictionary knowledge and Section 5.4 introduces our method of deriving global indicators. Section 5.5 shows the experimental results. Section 5.6 discusses the effectiveness of global indicators in identifying summary keywords that have low frequency in the input. We conclude this chapter in Section 5.7.

5.2 Related Work

The idea of using global knowledge dates back to Edmundson's seminal work on single-document extractive summarization (Edmundson, 1969). There, the author proposed four methods of weighting sentences: *Cue Method*, *Key Method*, *Title Method* and *Location Method*. According to Edmundson (1969), "*The Cue Method is based on the hypothesis that the relevance of a sentence is affected by the presence of pragmatic words (cue words).*" The cue words include *Bonus words* (which indicates a sentence including this word is likely to be selected) and *Stigma words* (which indicates a sentence including this word is unlikely to be selected), along with others. These words are derived based on the statistics of the words from 100 extract-article pairs, where the words that appear in the summaries with a high (low) selection ratio (defined as the number of occurrences in human extracts to the number of occurrences in the original articles) are identified as Bonus words (Stigma words). The weight of an input sentence is computed based on the number of Bonus and Stigma words in the sentence. Linguistic analysis shows that the bonus words include "*comparatives, superlatives, adverbs of conclusion, value terms, relative interrogatives, causality terms*", while the stigma words include "*anaphoric expressions, belitting expressions, insignificant-detail expressions, hedging expressions*".

The *Cue Method* makes use of the words that would trigger the selection or exclusion of a sentence in general. Hence, it is different from the *Key Method*, *Title Method* and *Location Method* he proposed, which all rely on characteristics of the input. Interestingly, the result of Edmundson suggests that the Cue Method performs better than word frequency (i.e., Key Method) for his task. Subsequent early papers also use cue words for summarization in different domains (Pollock and Zamora, 1975; Paice, 1980; Paice and Jones, 1993).

Global knowledge has also been used as features in early supervised single-document summarization systems. Kupiec et al., (1995) developed a trainable summarization program that predicts the inclusion of the input sentences. There, features are set to indicate if a sentence contains the phrases in a fixed phrase list (e.g., “this letter”, “in conclusion”). Teufel et al. (1997) replicated the design of Kupiec et al. (1995), where they use a more meticulously designed phrase list. The phrases are manually classified into five classes, which indicate “*the likelihood of a sentence containing a given clue (phrase) to be included in a summary*”. The sentences are then classified into these five classes, based on the classes of the phrases in the sentence. Features are set based on the class that one sentence belongs to.

The idea of identifying globally important words has also been utilized for multi-document summarization. Schiffman et al. (2002) suggested to identify globally important words (i.e., *lead word*) based on the appearance of a word in the lead sentences of a large corpus. Let $p_l(w)$, $p_i(w)$ denote the probability that word w appears in the lead sentences and anywhere in the input, respectively. w is regarded as a lead word if $r = p_l(w)/p_r(w) > 1$ and r is statistically significant by binomial test. The lead words are regarded as keywords and the importance of a sentence in the input is determined based on the number of lead words in the sentence.

In general, however, global knowledge has not been widely used in systems developed in the past 15 years. It is only within the past two years that we see a resurgence of this idea. Our paper (Hong and Nenkova, 2014a) employs dictionary knowledge and intrinsic word importance as features for content selection. Nye and Nenkova (2015) identify newsworthy verbs in the domain of world news from the NYT corpus. The newsworthy verbs are identified based on the change of probabilities between the human summary and the input, similar to Woodsend and Lapata (2012) and Hong and Nenkova (2014a). They show that verbs which tend to appear in summaries are more active, hostile, and negative; while the verbs which tend to appear in articles describe more personal actions. Li et al. (2015) leverage multiple external resources

(Wikipedia, WordNet, SentiWordNet, word embeddings) to improve the weighting of bigrams in order to produce better summaries for query-focused summarization. Two methods are used to incorporate global knowledge for bigrams. First, they adapt our method (Section 5.4) of estimating intrinsic importance for bigrams. Second, they use SentiWordNet (Esuli and Sebastiani, 2006) to indicate the sentiment score of a bigram, which resembles our use of MPQA (Section 5.3.1). The authors also set features to indicate whether a bigram is among the top-k most frequent bigrams from Wikipedia pages. However, since the Wikipedia pages are selected based on the current input, we regard this kind of external knowledge as partially input specific. A very recent paper (Cao et al., 2015b) applies an enhanced version of convolutional neural networks to generate summary prior features automatically.

Different from summarization, global knowledge (or broadly speaking, knowledge extracted from external resource) is widely used for keyphrase extraction (Hasan and Ng, 2014), where the task is to "*select important and topical phrases from the body of the document*" (Turney, 2000). Medelyan et al. (2009) show that phrases which appear more frequently in links in Wikipedia pages are more likely to be keyphrases. Query logs (Yih et al., 2006) and terminological databases (Lopez and Romary, 2010) are also leveraged to improve the performance of keyphrase extraction.

Our work is also related to the summarization approaches that rely on pre-defined templates. For example, the famous SUMMONS system (McKeown and Radev, 1995; Radev and McKeown, 1998) processes the input by filling in template slots, then generates the final summary based on these templates. The templates, created by experts, can be regarded as global knowledge of a specific domain.

5.3 Mining Global Knowledge from Dictionaries

We hypothesize that words in certain categories are likely to be globally important or unimportant. In fact, many summarizers already rely on this hypothesis for content selection. Systems that conduct rule-based sentence compression use part-of-speech (POS) tags to identify the words to be removed (Dunlavy et al., 2003; Conroy et al., 2006b; Wang et al., 2013). Event based summarizers rely on named entity (NE) recognition (Filatova and Hatzivassiloglou, 2004; Li et al., 2006) to identify important events. Moreover, our experiments in Section 3.5.2 have shown that words with certain POS or NE tags are likely or unlikely to appear in human summaries.

In this section, we use MPQA and LIWC dictionaries to build features for our word importance estimation model (Chapter 3). For each word, a feature has value 1 if the word belongs to the category that the feature corresponds to, otherwise 0. To test the predictive power of these features, we perform proportion test and Wilcoxon rank-sum (WRS) test (a.k.a Mann-Whitney U test) for the words that are used and not used in human summaries. Since the features are binary, *we regard proportion test as our main metric.*[2] Our experiments are helpful to understand what categories might be associated with the inclusion or exclusion of a word, independent of a particular input.

Using dictionaries to build features gives us two advantages. First, in contrast to the sparse unigram features, we have a dense representation of the feature space. Consider a word that appears in the test data but not in the training data. Because this word belongs to a dictionary category, we can infer the property of this word based on words of the same category in the training data. However, unigram features give no signals in this case. Second, the feature space is not determined based on a specific input. This is a more advantageous representation, according to Yang and Nenkova (2014).

[2] We observe that these two tests generate similar p-values and the relative rank between features are also very similar.

Features	Sample words	prop	WRS	+/-	r_f
strong subj & negative	fear, concerned, trouble, worst	**1e-4**	**1e-4**	-	8.5%
strong subj & neutral	air, opinion, felt, feel, view	0.057	**0.045**	-	9.3%
strong subj & positive	great, hope, true, kind, agree	**0.009**	**0.007**	-	7.3%
weak subj & negative	force, close, lost, hard, war, crisis	0.147	0.135	+	12.3%
weak subj & neutral	move, major, pressure, high, show	0.14	0.124	-	9.3%
weak subj & positive	clear, good, minister, deal, leading	0.973	0.939	-	10.3%

Table 5.1: The MPQA features and their p-values by proportion test (prop) and Wilcoxon rank-sum test (WRS). +/- indicates more in the summary/input. **Bold** indicates statistical significant ($p < 0.05$). r_f indicates the percentage of words with this feature tag in the input that appear in human summaries. The mean r_f for all words is 10.9%.

5.3.1 Multi-Perspective Question Answering (MPQA)

In the MPQA lexicon (Wiebe and Cardie, 2005), each word is labeled based on its subjectivity and polarity. There are two subjectivity (strongly subjective, weakly subjective) and three polarity (positive, neutral, negative) categories. The words that are subjective in most contexts are labeled as strongly subjective (e.g., abase, abash, abysmal), while the words that are subjective in some contexts are labeled as weakly subjective (e.g., abandon, ability, accept). Objective words are not included in this lexicon. The polarity label indicates whether or not one word evokes people's negative, neutral or positive emotions.

For each word, we construct six features; each feature corresponds to a combination of different polarities and subjectivities. Table 5.1 shows the p-value from significance test and sample words for each category. r_f is defined as the percentage of words with this feature tag in the input that appears in human summaries (for formal definition of r_f, see Section 3.5.2). Experiment shows that words with strong subjectivity, whether positive, neutral or negative, are less likely to be used in summaries. Most strikingly, the p-value for strongly subjective negative words is very low: about 10^{-4}.

There are two possible explanations towards why words with strong subjectivity are unlikely to be used in summaries: (1) the main topic of an input tends not to be too subjective, or (2) the abstractors tend not to be too subjective while writing summaries. To investigate which explanation is more plausible, we look into some examples. First, “feel” (strong subjectivity and neutral) has never appeared in human summaries and appears 26 times in 11 inputs. By looking at examples, we observe that almost all occurrences of this word is within quotations (see Table 5.2). Therefore, the exclusion of “feel” is due to the fact that quotations are unlikely to be included in summaries.[3] Second, “fear” (strong subjectivity and negative) has never appeared in human summaries and appears 26 times in 15 inputs. We observe that the usage of this word is mostly concerned with people expressing their feelings. This is consistent with an observation that verbs which describe personal actions are unlikely to appear in summaries (Nye and Nenkova, 2015). For both examples, the first explanation is more plausible, i.e., the strongly subjective words are not used because they are generally not related to the central topic of the input.

“I **feel** guilty”, he said, close to tears as he stood near the flowers and candles at the fire site.

The beatification may also owe its speed to the pope’s **fear** that a successor may be less sensitive to the East Europe an Church’s struggle against communism, to which he has devoted much of his life.

Table 5.2: Examples of input sentences that include words with strong subjectivities.

5.3.2 Linguistic Inquiry and Word Count (LIWC)

The LIWC application (Tausczik and Pennebaker, 2007) is originally designed to conduct text analysis for psychological research. Representative applications of LIWC

[3] Among the words that only appear in quotations of an input, 3.2% of them are used in human summaries. This is much smaller compared to the percentage for all words (10.9%).

Features	**Sample words**	**prop**	**WRS**	**+/-**	r_f
death	killed, killing, war, died, death	1.5e-13	6.5e-14	+	20.4%
anger	killed, killing, victims, hit, attack	3e-9	2.1e-9	+	21.7%
achieve	president, leader, leaders, control	1.9e-5	1.5e-5	+	13.3%
negative emotion	killed, killing, lost, pressure, victim	0.005	0.004	+	14.3%
money	business, economic, bill, economy	0.016	0.013	+	11.4%
inclusive words	including, included, close, open	0.023	0.016	+	13.7%
space	international, country, national, world	0.050	0.045	+	12.0%
perceptual process	spokesman, speaking, hand, press	3e-6	2.2e-6	-	4.4%
insight	statement, question, found, believed	5e-6	4.1e-6	-	5.7%
hear	spokesman, speaking, heard, spoke	5.4e-5	3.2e-5	-	1%
tentative	maybe, question, hope, appears	0.001	7e-4	-	4.0%
cognitive process	make, news, statement, including	0.002	2e-3	-	8.8%
present tense	make, give, carry, turn	0.004	0.003	-	5.4%
body	head, face, hand, feet, hands	0.005	0.004	-	5.1%
friend	friends, neighboring, neighbors, fellow	0.026	0.015	-	0.7%
function words	part, main, half, back	0.023	0.019	-	7.5%
positive emotions	great, good, important, support	0.044	0.039	-	8.4%

Table 5.3: Significant LIWC features and their p-values by proportion test (prop) and Wilcoxon rank-sum test (WRS). +/- indicates more in the summary/input. r_f indicates the percentage of words with this feature tag in the input that are included in human summaries. The mean r_f for all words is 10.9%.

include suicide risk assessment (Matykiewicz et al., 2009), deception detection (Newman et al., 2003), sentiment analysis (Tumasjan et al., 2010), and schizophrenia identification (Hong et al., 2012; Hong et al., 2015b). The center of LIWC is a dictionary that assigns tags to words based on their lexical or semantic properties, which makes dictionary is appropriate for our analysis. One drawback of the LIWC dictionary lies in its low coverage, which only includes 4500 words or word-stems. It would be interesting if similar analysis can be conducted based on dictionaries with a higher coverage for future work.

Table 5.1 shows the p-value by Wilcoxon rank-sum (WRS) test and proportion test. Among all 64 features that correspond to LIWC categories, 16 are significant by proportion test, 17 are significant by WRS test.

Categories that appear at a higher rate in human summaries (i.e., summary-biased categories) include *death*, *anger*, *achievements*, *negative emotions*, *money*, *inclusive words* and *space*. Among which, the first two are extremely significant. Indeed, words that are related to death (e.g., killed, killing, war, died), anger (e.g., killed, killing, attack, victims) are the main focus of many news articles. The statistical significance of negative emotions is related to a finding in prior work that general sentences (which are likely to be summary sentences) include a greater number of polarity words (Louis and Nenkova, 2011a; Louis and Nenkova, 2011b).

“Killed” and ”killing” are the most frequent words in the categories *death*, *anger* and *negative emotions*. Therefore, it is possible that these categories are significant because these two words appear too often. To test if it is the case, we remove the samples that correspond to “killed” and “killing” in our data. Then we retest the statistical significance of these features. The new p-value for death and anger are still very significant—smaller than 0.002 by proportion test. However, the new p-value for negative emotion is 0.118. The result of WRS test is also very similar.

A greater number of features are concerned with the categories that are unlikely to be used in human summaries (i.e., input-biased categories). Highly significant categories include *perceptual process* (spokesman, speaking, hand), *insight* (statement, question) and *friend* (friends, neighboring, fellow). In LIWC, words are also grouped together by their part-of-speech tags, where we observe less occurrences of present tense verbs and function words in human summaries.

5.4 Estimating Intrinsic Importance of Words

This section investigates the hypothesis that people have intrinsic preference towards the inclusion or exclusion of certain words. In Section 5.4.1, we propose two methods to compute the intrinsic importance of words (i.e., *global indicators*). In order to better understand why some words are globally important/unimportant, we analyze the categories that these words belong to in Section 5.4.2. Section 5.4.3 presents a system only based on the global indicators, which verifies the effectiveness of these indicators. Section 5.4.4 describes how the global indicators are used as features for summary keyword identification.

5.4.1 Deriving the Global Indicators

Below we introduce two methods of computing global indicators, where the second method is an ungraded version of the first one. The indicators are computed based on $160,001$ summary-article pairs from year 2004 to year 2007 of the New York Times corpus. The idea is to compute the change of probability of each word between the summary and the original article. Similar approach has been used in Woodsend and Lapata (2012) for identifying words that are likely to be avoided in summaries. However, their analysis is based on a very small corpus (i.e., one TAC dataset). Gillick and Dunietz (2014) utilize this corpora for generating a labeled corpus of entity salience identification.

Method 1

We build two unigram language models (LMs): one from the original articles (LM_G), the other from the summaries (LM_S). Here we use the SRI Language Modeling Toolkit (SRILM) (Stolcke, 2002) with Ney's absolute discounting (Ney et al., 1994) and 0.75 as the constant to subtract.[4] Since SRILM uses white space as the word separator, we tokenize all files by Stanford CoreNLP (Manning et al., 2014) before building the LMs. The probability of word w in LM_S and LM_G are denoted as $P_S(w)$ and $P_G(w)$ respectively.

We compute the intrinsic importance (global indicators) of w based on $P_S(w)$ and $P_G(w)$. As the corpus is large enough, we only consider the words that appear in both at least one article and one summary. This results in a total of $128,381$

[4] ./ngram-count with parameters: -cdiscount 0.75

words. We introduce *five different global indicators*. $Score_1(w)$ is the probability of w in the human summaries. Intuitively, words that appear more often in summaries are likely to be important. $Score_2(w)$ ($Score_3(w)$) computes the difference (ratio) between $P_S(w)$ and $P_G(w)$. Moreover, we compute $Score_4(w)$, where the formula resembles Kullback-Leibler (KL) divergence. This is based on the hypothesis that summary-biased words tend to have higher $P_S(w)$ and larger difference/ratio between $P_S(w)$ and $P_G(w)$. Similarly, we compute $Score_5(w)$ to characterize the unimportant words (input-biased). $Score_4(w)$ and $Score_5(w)$ are regarded as our main metrics.

$$\begin{aligned}
Score_1(w) &= P_S(w) && (5.1)\\
Score_2(w) &= P_S(w) - P_G(w) && (5.2)\\
Score_3(w) &= P_S(w)/P_G(w) && (5.3)\\
Score_4(w) &= P_S(w) \cdot \ln \frac{P_S(w)}{P_G(w)} && (5.4)\\
Score_5(w) &= P_G(w) \cdot \ln \frac{P_G(w)}{P_S(w)} && (5.5)
\end{aligned}$$

Table 5.4 shows the top words and top content words, ranked by the five types of global indicators. Here we briefly discuss the *content words*. Words that tend to be used in summaries, characterized by high $Score_4(w)$, include locations (e.g., York, NJ, Iraq), people's names (e.g., Bush, John), abbreviations (pres, corp, dept) and verbs of conflict (e.g., contends, dies). Some of the top ranked words, such as Iraq, Bush and John, are related to the big events happened between 2004 and 2007. Words that tend not to be used in human summaries, characterized by high $Score_5(w)$, include courtesy titles (e.g., Mr, Ms, Jr.), relative time reference (e.g., yesterday, p.m., Tuesday) and verbs that people use to express opinions (e.g., asked, told, added). The words with high probability in summaries ($Score_1(w)$) overlap with those ranked high by $Score_4(w)$ to some extent, but also includes a number of frequent words that appear often both in the summaries and in the original articles (e.g. State, million, American, percent). The words ranked high by $P_S(w) - P_G(w)$

($Score_2(w)$) resembles that of $Score_4(w)$. The words ranked high by $P_S(w)/P_G(w)$ ($Score_3(w)$) include many abbreviations and uncommon words.

In summary, the global indicators—especially $Score_1(w)$, $Score_2(w)$, $Score_4(w)$ and $Score_5(w)$—seem to correlate well with our intuitions.

Method 2

Even though Method 1 successfully identifies the intrinsically important words, a scrutiny of Table 5.4 reveals two problems. First, many words are ranked high because of journalistic conventions (e.g., reviews, op-ed, correction, photo(s)). For example, the summary of a correction article includes the word "correction", the summary of an article accompanied by photos contains the word "photo(s)". These words do not describe the main topics of an article. Second, abstractors uses abbreviations a lot in summaries: "pres" for "president", "sen" for "senator", "min" for "minister", etc. As a result, many abbreviations are ranked undesirably high. Method 2 tackles these problems.

Let n denote the number of summary-article pairs. Let T_i ($1 \leq i \leq n$) denote the articles and let S_i ($1 \leq i \leq n$) denote the summaries. Method 2 includes the following steps:

Step 1: We manually build a dictionary, which includes words and their abbreviations (e.g., president—pres). This dictionary is included in Appendix B. For the occurrences of the abbreviations in S_i and T_i, we replace them with the original words. This helps to alleviate the second problem.

Step 2: For each S_i, we form a new summary S'_i by filtering out the words that have never appeared in its corresponding original article (T_i). This helps to tackle the first problem. Moreover, this step is more suitable for extractive summarization (which is our case), because the words not in the original article cannot be selected by extractive summarizers.

Metric	Rank	Words
$P_S(w)$	1-8	of, to, and, in, m, that, on, for
	9-16	's, is, by, photo, new, with, at, from
	17-24	are, as, says, has, photos, s, who, will
	25-30	article, his, york, be, not, have
$P_S(w) - P_G(w)$	1-8	m, of, photo, says, new, on, photos, s
	9-16	in, by, article, to, column, york, letter
$P_S(w)/P_G(w)$	1-8	atty, pres, fda, region/long, aclu, irs, guantanamo, faa
	9-16	nj, nfc, nc, dept, chairman-chief, region/new, dist
$P_S(w) \cdot \ln \frac{P_S(w)}{P_G(w)}$	1-8	m, photo, photos, pres, says, article, column
	9-16	of, reviews, letter, new, on, by, york, in
	17-24	in, l, sen, ny, discusses, drawing, to, op-ed, holds
	25-30	correction, bush, editorial, and, j, will
$P_G(w) \cdot \ln \frac{P_G(w)}{P_S(w)}$	1-8	the, a, mr., said, an, i, n't, he
	9-16	you, was, we, ms, it, had, this, but
	17-24	she, 're, my, yesterday, here, like, they, were
	25-30	me, 've, there, do, 'm, so

Metric	Rank	Words
$P_S(w)$	1-8	photo, photos, article, york, column, letter, bush
	9-16	state, reviews, million, american, pres, percent, iraq, years
	17-24	people, government, year, john, company, correction, national
	25-30	federal, officials, drawing, billion, public, world, administration
$P_S(w) - P_G(w)$	1-8	photo, photos, article, column, york, letter, reviews, pres
	9-15	bush, city, state, correction, drawing, op-ed, iraq
$P_S(w)/P_G(w)$	1-8	atty, pres, fda, region/long, aclu, irs, guantanamo, faa
	9-15	nj, nfc, nc, dept, chairman-chief, region/new, dist
$P_S(w) \cdot \ln \frac{P_S(w)}{P_G(w)}$	1-8	photo, photos, pres, article, column, reviews, letter, york
	9-16	sen, ny, discusses, drawing, op-ed, holds, correction, bush
	17-24	editorial, dept, city, nj, min, map, corp, graph
	25-30	contends, iraq, john, dies, sec, state
$P_G(w) \cdot \ln \frac{P_G(w)}{P_S(w)}$	1-8	mr, ms, yesterday, p.m., lot, tuesday, ca, thursday
	9-16	wednesday, friday, told, monday, time, added, thing, sunday
	17-24	things, asked, good, night, saturday, nyt, back, senator
	25-30	wanted, kind, jr., mrs , bit, looked

Table 5.4: Top words derived by five global importance estimation methods (Method 1). The top table includes all words, the bottom table includes content words only. All words are lowercased.

Metric	Rank	Words
$P_S(w)$	1-8	of, to, and, in, that, on, 's
	9-16	is, by, new, with, at, as, from, are
	17-24	has, who, will, his, be, not, have, it
	25-30	york, he, about, the, was, its
$P_S(w) - P_G(w)$	1-8	of, to, in, and, new, on, by, for
	9-15	york, 's, is, will, bush, that, president
$P_S(w)/P_G(w)$	1-7	perval, bacteriophages, juppe, melby, raveche, inderfurth, tikshoret
	8-14	friedman-simring, lavoung, aclu, korondi, mckiver, gronim, meini
$P_S(w) \cdot \ln \frac{P_S(w)}{P_G(w)}$	1-8	of, in, to, new, and, on, by, m.
	9-16	york, for, bush, article, president, will, city, is
	17-24	's, are, state, iraq, that, million, from, john
	25-30	editorial, billion, at, federal, percent, has
$P_G(w) \cdot \ln \frac{P_G(w)}{P_S(w)}$	1-8	the, a, mr., said, an, i, we, n't
	9-16	you, ms., he, it, was, had, this, she
	17-24	my, but, 're, yesterday, here, there, me, like
	25-30	they, 've, do, our, were, 'm

Metric	Rank	Words
$P_S(w)$	1-8	york, president, city, bush, state, million, percent, american
	9-16	years, iraq, company, government, article, people, year, john
	17-24	federal, national, billion, officials, public, administration, world
	25-30	united, court, group, house, police, war, school
$P_S(w) - P_G(w)$	1-8	york, bush, president, city, state, million, iraq
	9-15	john, percent, american, billion, government, federal, administration
$P_S(w)/P_G(w)$	1-7	perval, bacteriophages, juppe, melby, raveche, inderfurth, tikshoret
	8-14	friedman-simring, lavoung, aclu, korondi, mckiver, gronim, meini
$P_S(w) \cdot \ln \frac{P_S(w)}{P_G(w)}$	1-9	york, bush, article, president, city, state, iraq, million, john
	10-16	editorial, billion, federal, percent, american, government
	17-22	administration, jersey, michael, national, column, court, senator
	23-30	company, op-ed, gov., security, department, minister, directed, police
$P_G(w) \cdot \ln \frac{P_G(w)}{P_S(w)}$	1-8	mr., ms., yesterday, sept., ca, lot, tuesday, thursday
	9-16	n.y., wednesday, friday, told, monday, thing, added, things
	17-24	time, nyt, asked, good, night, p.m., mrs., sunday
	25-30	saturday, wanted, back, thought, looked, wo

Table 5.5: Top words derived by five global importance estimation methods (Method 2). The top table includes all words, the bottom table includes content words only. All words are lowercased.

Step 3: This step builds the language models. We first build a language model (LM) from all texts T_i, using the same approach as described in Method 1. We then build a LM for all S'_i, where the words from T_i are used as the full vocabulary list.[5] The vocabulary list includes $587,976$ words, which is at least four times as large as the word list in Method 1. Compared to Method 1, words that appear in T_i but not in S'_i are considered for Method 2.

Step 4: The following steps are the same as Method 1: we calculate $Score_1(w), \ldots,$ $Score_5(w)$ to quantify the intrinsic word importance.

Table 5.5 includes the top words ranked by $Score_1(w)$, $\ldots$, $Score_5(w)$ using Method 2. Examination shows that it include less noisy information compared to Table 5.4. For instance, among the words that are newly promoted to the top 30 ranked by $Score_4(w)$, we see "government", "administration", "senator", "minister" and "police", which are often used to characterize the main events of a news article.

5.4.2 Analysis based on Dictionaries

So far, we have proposed methods of identifying summary-biased and input-biased words. The reason why these words get identified might be because they evoke topics (i.e., categories of words) that are intrinsically important to humans. To capture this, we analyze the categories that these words belong to based on dictionaries (i.e, MPQA and LIWC). Our experiments are inspired by the study of Nye and Nenkova (2015), where they analyze verbs in the domain of world news. The General Inquirer Lexicon (Stone et al., 1966) is used in their analysis.

Formally, let W_S and W_I denote the top $10K$ words ranked by $Score_4(w)$ (summary-biased) and $Score_5(w)$ (input-biased), respectively. Let W_{C_S} ($W_{C_S} \in W_S$) and W_{C_I} ($W_{C_I} \in W_I$) denote the summary-biased and input-biased words with the category tag C, respectively. We are interested in the categories that include the highest

[5] ./ngram-count with parameters: -cdiscount 0.75 -vocab vocabulary.txt

number of summary-biased and input-biased words. Moreover, for each category C, we compute $|W_{C_S}|$ and $|W_{C_I}|$ (i.e., cardinality of the two sets). Suppose $|W_{C_S}|$ is much larger than $|W_{C_I}|$ for C, then it suggests that words with this tag are more likely to be regarded as intrinsically important rather than unimportant.

Analysis based on MPQA categories

Figure 5.1 shows the number of summary-biased and input-biased words for each subjectivity & polarity tag in MPQA. The top two categories for summary-biased words are *weakly subjective & negative* and *weakly subjective & positive*; while the top two categories for the input-biased words are *strong subjective & negative* and *strong subjective & positive*. Especially, for *weakly subjective & negative (positive)* words, $|W_{C_S}|$ is larger than $|W_{C_I}|$; while for *strongly subjective & negative (positive)* words, $|W_{C_S}|$ is much smaller than $|W_{C_I}|$. Clearly, the global importance of words are related to subjectivities of these words. Our result is also consistent with our observation in Section 5.3.1 that humans tend to avoid the words that are strongly subjective based on the smaller DUC set.

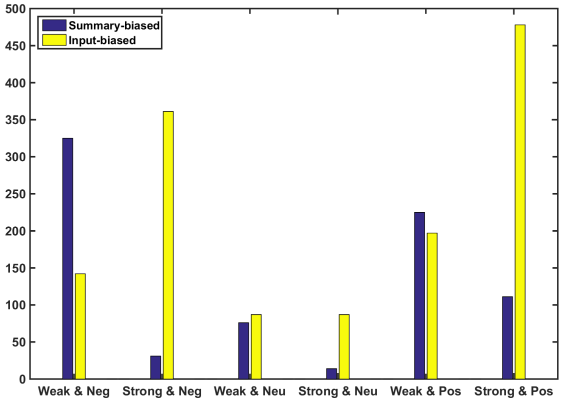


Figure 5.1: The number of summary-biased and input-biased words that belong to one of our six MPQA categories. Strong/Weak is short for Strongly/Weakly subjective. Pos is short for positive, Neg is short for negative, Neu is short for neutral.

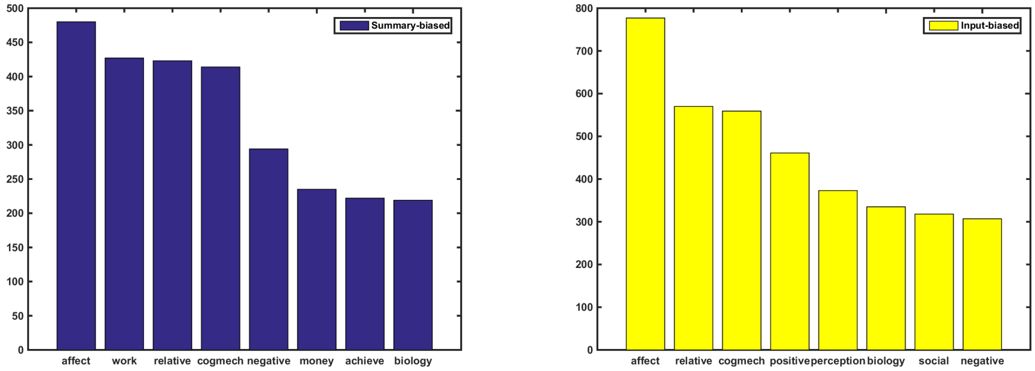


Figure 5.2: The top eight LIWC categories that include the highest number of summary-biased (left figure) and input-biased (right figure) words.

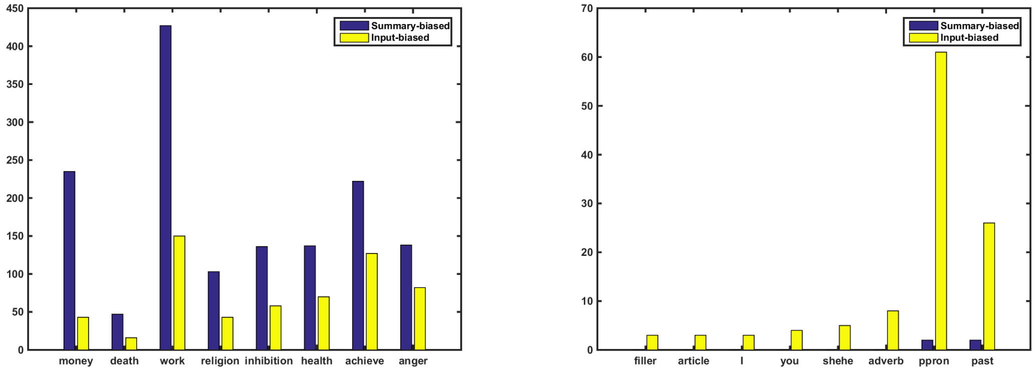


Figure 5.3: The top eight summary-biased (sorted by $|W_{C_S}|/|W_{C_I}|$, left figure) and input-biased (sorted by $|W_{C_I}|/|W_{C_S}|$, right figure) LIWC categories.

Analysis based on LIWC categories

Figure 5.2 shows the top eight LIWC categories that include the highest number of summary-biased and input-biased words. The top eight categories for summary-biased words are *affective process*, *work*, *relativity*, *cognitive process*, *negative words*, *money*, *achievement* and *biological process*. The top eight categories for input-biased words are *affective process*, *relativity*, *cognitive process*, *positive words*, *perceptual process*, *biological process*, *social process* and *negative words*.[6] Interestingly, *negative*

[6] Here are sample words in some of the categories. *affective process*: "happy", "cried", "abandon"; *relativity*: "area", "bend", "exit", "stop"; *cognitive process*: "cause", "know", "ought";

words is the fifth highest ranking category for summary-biased words, and it is ranked the eighth for input-biased words. However, the absolute number of $|W_{C_S}|$ and $|W_{C_I}|$ are similar. Thus it is somewhat unclear if our conclusion that negative words are more summary-biased on the DUC corpus holds for the NYT corpus. Negative verbs are likely to be more summary-biased for world news (Nye and Nenkova, 2015).

In order to understand which semantic categories can better characterize summary and input biased words, we show the top eight summary-biased (ranked by $|W_{C_S}|/|W_{C_I}|$) and input-biased (ranked by $|W_{C_I}|/|W_{C_S}|$) categories in Figure 5.3. Words with tags *money*, *death*, *religion*, *inhibition*, *health*, *achieve* and *anger* are likely to be included in summaries.[7] Many of these categories are also significant based our analysis on the DUC dataset (see Table 5.3). We also observe humans tend to avoid some function words (e.g. the categories *article*, *I*, *heshe* and *pronoun* in LIWC) and adverbs while writing summaries.

It is worth noting that there are less *past tense verbs* in human summaries of the NYT corpora (Figure 5.3, right figure). In contrast, we find that there are less *present tense verbs* and more *past tense verbs* in human summaries of the DUC 2003 data (see Table 3.3). Our results suggest that there might be stylistic differences between the two datasets. Indeed, by looking at sample summaries from the two datasets, we notice that human summaries in the DUC 03 dataset are mostly written in past tense forms; while human summaries in the NYT corpora are sometimes written in present tense forms.

5.4.3 Blind Sentence Extraction

In Section 5.4.4, we will use the estimates of intrinsic word importance (i.e., global indicators) as features in our regression model that identifies summary keywords.

biological process: "eat", "blood", "pain"; *perceptual process*: "observing", "heard", "feeling". Examples from: `http://www.liwc.net/descriptiontable1.php`

[7] Here are the sample words in some of the categories. *religion*: "altar", "mosque", "church"; *health*: "clinic", "flu", "pill"; *achieve*: "earn", "hero", "win"; *anger*: "hate", "kill", "annoyed".

Before turning to that, however, we confirm the effectiveness of the global indicators. We present a summarization system: *Blind*, which generate summaries solely based on the global indicators. We show that by using these general indicators that approximate human's preference, Blind can produce summaries of decent quality.

Formally, let I denote the input, we rank all content words from I by $Score_4(w)$ $(P_S(w) \cdot \ln \frac{P_S(w)}{P_G(w)})$. The top k words are assigned weight 1, the others are assigned weight 0. Based on these binary weights, Blind estimates sentences in the input and generates summaries based on GreedySum (described in Section 4.1.2). Figure 5.4 shows the performance of Blind, with $Score_4(w)$ estimated by Method 1 and Method 2, respectively. The performance is stable when $220 \leq k \leq 500$, with both systems achieving a ROUGE-1 (R-1) of at least 0.323 and a ROUGE-2 (R-2) of at least 0.052. Method 2 has a better performance than Method 1 in terms of R-2.

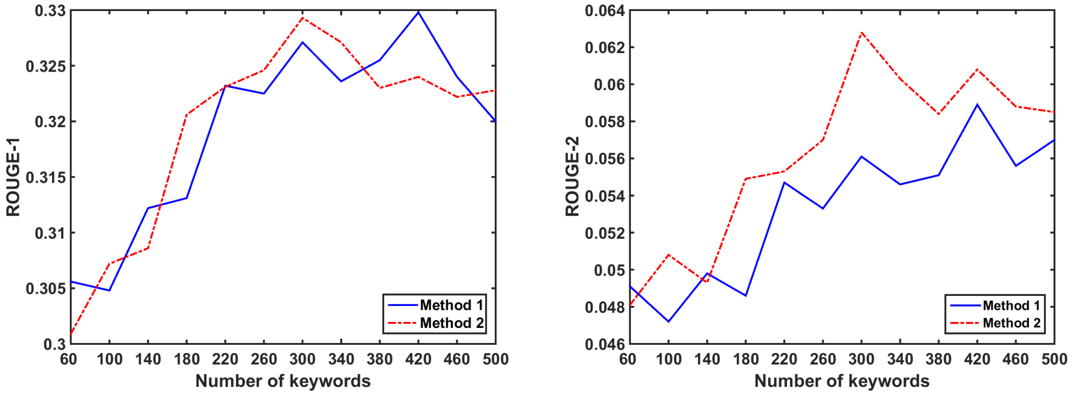


Figure 5.4: ROUGE-1 (left) and ROUGE-2 (right) of the Blind summarizers on the DUC 2004 dataset. x-axis shows the number of words that are regarded as keywords.

Table 5.6 compares the performance of Blind with three baselines: Random, LatestLead and FirstSent. k is set to the value where Method 1 and Method 2 achieve the peak performance in terms of R-2, respectively. Our systems perform significantly better (two-sided paired Wilcoxon test, same below) than picking sentences from the input randomly (Random). We also achieve a similar R-2 and a higher R-1 compared to LastLead, which selects the first 100 words from the latest article in the input. However, our systems perform significantly worse than FirstSent, formed by selecting the first sentences from each article of the input. Table 5.7 shows the sample summaries generated by Random, Blind (Method 1), Blind (Method 2) and Firstsent. It can be easily seen that Blind generates better summaries than Random.

In summary, our results confirm that the global indicators encode information that is highly indicative of informative content.

Systems	ROUGE-1	ROUGE-2	ROUGE-4
Random	0.3032	0.0442	0.0036
Blind (Method 1) (420 keywords)	0.3298	0.0589	0.0073
Blind (Method 2) (300 keywords)	0.3293	0.0628	0.0078
LatestLead	0.3139	0.0611	0.0063
FirstSent	0.3426	0.0722	0.0121

Table 5.6: Performance of the Blind systems and three baseline systems: Random (randomly selecting sentences), LatestLead (using the first L words of the last article) and FirstSent (using the first sentences from each article of the input).

5.4.4 Applying the Global Indicators

Our global indicators are directly used as features for two tasks: summary keyword identification (Chapter 3) and summary combination (Chapter 6). The indicators are derived by Method 1 for the first task, by Method 2 for the second task.

For the first task, the features are designed as follows: for each word, features are set to determine whether it is ranked within the top-k or bottom-k words (one feature for the top-k, one feature for the bottom-k), ranked by one of the five scoring functions $Score_1(w), \ldots, Score_5(w)$. Here k is selected from a set of pre-defined values.[8] Among all 70 features in this category, 52 of them are significant ($p < 0.05$) by proportion test, 48 of them are highly significant ($p < 0.001$). We will describe how the indicators are used for summary combination in Chapter 6.

[8] The values are: 100, 200, 500, 1000, 2000, 5000, 10000.

Random Summary

It was sunny and about 14 degrees C (57 degrees F) in Tashkent on Sunday. The president is a strong person, and he has been through far more difficult political situations, Mityukov said, according to Interfax. But Yeltsin's aides say his first term, from 1991 to 1996, does not count because it began six months before the Soviet Union collapsed and before the current constitution took effect. He must stay in bed like any other person, Yakushkin said. The issue was controversial earlier this year when Yeltsin refused to spell out his intentions and his aides insisted he had the legal right to seek re-election.

Summary from the Blind system (Method 1, k = 420)

Russia's constitutional court opened hearings Thursday on whether Boris Yeltsin can seek a third term. Yeltsin's growing health problems would also seem to rule out another election campaign. The Russian constitution has a two-term limit for presidents. Russian president Boris Yeltsin cut short a trip to Central Asia on Monday due to a respiratory infection that revived questions about his overall health and ability to lead Russia through a sustained economic crisis. The upper house of parliament was busy voting on a motion saying he should resign. The start of the meeting was shown on Russian television.

Summary from the Blind system (Method 2, k = 300)

Russian President Boris Yeltsin cut short a trip to Central Asia on Monday due to a respiratory infection that revived questions about his overall health and ability to lead Russia through a sustained economic crisis. In the Kremlin, Yeltsin discussed developments in Kosovo with Primakov, Foreign Minister Igor Ivanov and Defense Minister Igor Sergeyev, Russian news agencies said. Yeltsin's premature return to Moscow also prompted doubts about his capacity to respond decisively in the Kosovo crisis, in which Russia has been leading a campaign to forestall airstrikes. Whenever Yeltsin falls ill, speculation arises about his ability to govern. The two sides are expected to sign economic agreements Monday.

Summary that only includes the first sentences from the input

President Boris Yeltsin has suffered minor burns on his right hand, his press office said Thursday. President Boris Yeltsin's doctors have pronounced his health more or less normal, his wife Naina said in an interview published Wednesday. President Boris Yeltsin, on his first trip out of Russia since this spring, canceled a welcoming ceremony in Uzbekistan on Sunday because he wasn't feeling well, his spokesman said. Doctors ordered Russian President Boris Yeltsin to cut short his Central Asian trip because of a respiratory infection and he agreed to return home Monday, a day earlier than planned, officials said.

Table 5.7: Summaries generated by Random (randomly selecting sentences), Blind, and FirstSent (using the first sentences from each article of the input).

5.5 Experiments and Results

This section studies how the removal of global knowledge features affects the performance of summary keyword identification and summarization.

Table 5.8 shows the performance of our models in identifying words that appear in at least k human summaries (denoted as G_k). This table overlaps with Table 3.6 to some extent: Table 5.8 reports precision, recall and F_1-score, while Table 3.6 reports the Keyword Pyramid Score and F_1-score. Two-sided Wilcoxon signed-rank test is used to compare the performance between RegSum and the models after ablation.

	100 words vs G_1			35 words vs G_2			15 words vs G_3			6 words vs G_4		
	P	R	F	P	R	F	P	R	F	P	R	F
RegSum (all)	.485	.451	.466	.499	.514	.503	.524	.530	.518	.493	.514	.479
-MPQA	.486	.452	.467	.495	.510	.500	.531	.536†	.524†	.500	.516	.484
-LIWC	.480	.446	.461	.494†	.508	.498	.529	.533	.522	.487	.506	.473
-NYT	.481	.448	.463	.496	.511	.500	.528	.532	.521	.470†	**.490**	**.456**
-All global	.483	.449	.464	**.490**	.505†	.494†	.532	.535	.525	.473†	.493†	.460
Method 2	.487	.453	.468	**.490**	**.505**	**.494**	.519	.523	.512	.483	.501	.468

Table 5.8: Performance of different feature classes in identifying summary keywords, evaluated on the DUC 2004 dataset. P, R, F = Precision, Recall, F_1-score. **Bold** indicates significantly different from RegSum ($p < 0.05$). † indicates that the difference is close to significant ($0.05 \leq p < 0.1$).

The MPQA features are ineffective; removing them sometimes leads to an increase in performance (e.g., when G_3 or G_4 is used as the gold-standard). Ablating LIWC features leads to a small decrease in general, except for the case when G_3 is used as the gold-standard. Ablating the global indicator (NYT) features also results in a small decrease most of the times, where the differences are significant at G_4. Finally, we ablate all global knowledge related features: this leads to a noticeable decrease in performance when G_2 and G_4 are used as the gold standards.

In general, removing global knowledge only results in a small decrease in performance. One possible explanation is that some of the removed features are correlated to the ones that are still in the feature set. For example, some LIWC features are in fact part-of-speech (POS) tags, which overlap with the POS features. Another explanation is that the predictive power of these features, though many of them are significant, are not as high as the more traditional ones (e.g., frequency, location).

In the original model, Method 1 is used to estimate word importance. We also experiment with using Method 2. Though Method 2 performs better than Method 1 when the weights are used in the Blind summarizer (see Figure 5.4), it does not perform as well as Method 1 when they are used as features for our task.

Finally, we apply the estimates of word importance for summarization, based on GreedySum described in Section 4.1.2. Table 5.9 shows the performance with each class of features removed. Removing all features related to global knowledge leads to a marginal decrease in performance on ROUGE-1, ROUGE-2, and ROUGE-4, while we can only observe a tiny change in performance after removing a single class of features. Using Method 2 does not help in generating better summaries.

	ROUGE-1	ROUGE-2	ROUGE-4
RegSum (all)	0.3875	0.0983	0.0150
-MPQA	0.3858	0.0979	0.0143
-LIWC	0.3877	0.0976	0.0150
-NYT	0.3858	0.0974	0.0145
-All global	0.3854	0.0959	0.0136
Method 2	0.3865	0.0976	0.0150

Table 5.9: Performance comparison between different feature classes on generic summarization, evaluated on the DUC 2004 dataset. Method 2 is different from Method 1 in the way of deriving global indicators.

5.6 Identifying Summary Keywords with Low Frequency in the Input

We test the hypothesis: global indicators are more helpful in identifying *summary keywords* that have low frequency in the input, compared to high frequency ones. Here *summary keywords* are defined as the words in the input that have appeared in the human summaries. We have this hypothesis, because:

- The frequency of a word in the input is highly indicative of whether or not one word should be used in human summaries. For high frequency words in the input, word frequency already give strong signals to suggest the inclusion of these words. However, it is difficult to decide whether or not a word should be included when it has low frequency. In this scenario, we hypothesize global knowledge to be more helpful.

Below we test our hypothesis by experiments. We regard the words that appear more than *five times* in the input as high frequency words, the others as low frequency words. Let H_I and L_I denote the high and low frequency words of the input I. Let G_{H_I} and G_{L_I} be the high and low frequency summary keywords (i.e., our gold standards), we have $G_{H_I} \subseteq H_I$ and $G_{L_I} \subseteq L_I$. The task is to predict the words in G_{H_I} (G_{L_I}) from H_I (L_I). Table 5.10 shows the mean cardinality of the sets H_I, L_I, G_{H_I}, G_{L_I} on the DUC 2003 and 2004 datasets.

	Avg. $\lvert H_I \rvert$	Avg. $\lvert L_I \rvert$	Avg. $\lvert G_{H_I} \rvert$	Avg. $\lvert G_{L_I} \rvert$
DUC 2003	120	981	49	59
DUC 2004	126	1014	50	59

Table 5.10: The average number of high ($|H_I|$) and low ($|L_I|$) frequency words per input, as well as the average number of high ($|G_{H_I}|$) and low ($|G_{L_I}|$) frequency summary keywords per input.

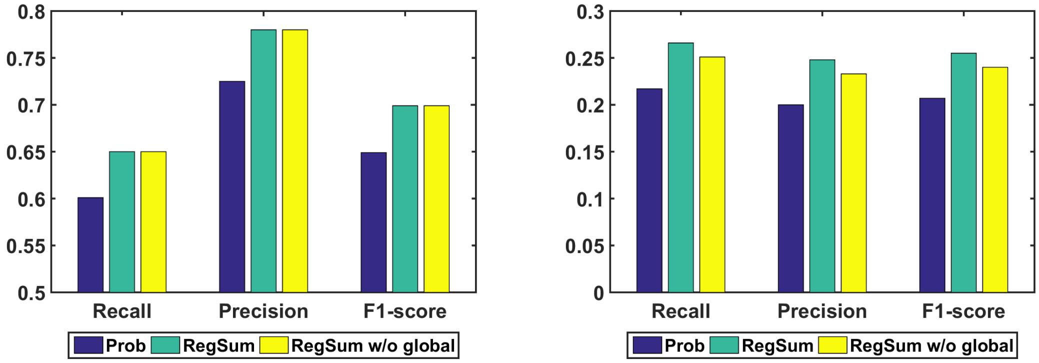


Figure 5.5: The performance of Prob, RegSum, and Regsum without global indicators in identifying summary keywords that appear with high/low (left/right) frequency in the input.

To rank the words in H_I and L_I, we compare three methods. First, we rank the words by their probabilities (Prob). If two words have same probability, the one with a larger LLR score is assigned a higher rank. Second, the words are ranked by their weights estimated by RegSum, where all features are used. Third, we ablate the global indicator features from RegSum, then train the model based on the rest of the features. After the words have been ranked, the top k_1 (k_2) words in H_I (L_I) are regarded as the predicted high (low) frequency summary keywords. Here we set $k_1 = 50$ and $k_2 = 60$, based on the mean of $|H_I|$ and $|L_I|$ on the DUC 2003 dataset.

Figure 5.5 compares the performance of these three methods. Apparently, identifying high frequency summary keywords is easier than identifying low frequency ones. We also observe that the two supervised methods perform much better than word probability. We then compare these two supervised methods. After ablating global indicator features, the performance decreased by 0.015 in identifying low frequency summary keywords, evaluated by precision, recall and F_1-score. The differences are all very significant by two-sided Wilcoxon signed-rank test ($p < 0.001$). However, global indicator features are not that helpful in identifying high frequency summary keywords.

In summary, if the words are of low frequency in the input, then global indicators are very helpful to improve the prediction of whether one word should be included in summary.

5.7 Conclusion

This chapter investigates the use of global knowledge for content selection in multi-document summarization. We propose two ways of mining global knowledge and test their effectiveness on various tasks related to content selection.

The first kind of global knowledge is mined from dictionaries. This is based on the hypothesis that categorical information (which can be mined from dictionaries) can capture general preference of humans towards certain topics, which is likely to be independent of input. Our analyses based on MPQA show that words with strong subjectivity are unlikely to be used in summaries. Our analyses based on LIWC reveal that topics such as *death*, *anger*, *negative emotions*, and *money* are likely to appear in summaries, while categories such as *perceptual process*, *hear*, *friends* and *function words* are unlikely to appear in summaries.

Second, we directly estimate the intrinsic importance of words (i.e., global indicators) to humans by analyzing summary-article pairs of news articles. We show that words such as “Bush”, “president”, “Iraq” are of high importance; while the words such as “Mr.”, “yesterday” and “added” are of low importance. We show that a system which selects sentences simply based on these indicators is able to generate summaries better than random and comparable to a standard baseline that selects the first L words from the most recent input document.

We have verified the effectiveness of global knowledge on four tasks. Ablating global knowledge leads to a small decrease in identifying summary keywords, summarization, and system combination (we will show this in Chapter 6). Notably, ablating global knowledge leads to a significant decrease in identifying summary keywords that have low frequency in the input.

In summary, our research has confirmed the usefulness of incorporating global knowledge for content selection. However, since the improvement is not large, how to better mine and employ global knowledge remains an interesting question. One very recent paper (Li et al., 2015) extends our method of estimating intrinsic word importance for bigrams and achieves promising results. Other methods of mining global knowledge is also productively explored in their work. Future research may focus on estimating content importance on various kinds of concepts.

Chapter 6

System Combination for Multi-document Summarization

6.1 Introduction

Research[1] in automatic summarization has seen great development in the past few decades. A great number of systems have been developed, using markedly different approaches. As we have shown in Section 4.2.4, the summaries generated by different systems are very diverse—the degree of overlap between the summaries from two systems is low in terms of words, sentences and summary content units (SCUs) (the definition of SCU can be found in Section 2.2.1). This suggests that combining summaries from different systems might be helpful in improving content quality.

A handful of papers have studied system combination for summarization. Based on the ranks of the input sentences assigned by different systems (i.e., *basic systems*), methods have been proposed to rerank these sentences (Wang and Li, 2012; Pei et al., 2012). However, these methods require the basic systems to assign importance scores to all input sentences. Thapar et al. (2006) combine the summaries from different

[1] This chapter is extended based on Hong et al. (2015a).

systems, based on a graph-based measure that computes summary-input/summary-summary similarity. However, their method does not show an advantage over the basic systems. In summary, few prior papers have successfully generating better summaries by combining the summaries from different systems (i.e., *basic summaries*).

In this chapter, we focus on *practical system combination*, where we combine the summaries generated by four portable unsupervised systems. We choose these systems, because: First, these systems are either off-the-shelf or easy-to-implement. Therefore, our method can be easily used on any new input sets. Second, though many methods have been proposed for summarization, the output of them are often available only on one dataset or even unavailable. Our setting enables us to conduct a large scale evaluation. Third, compared to more sophisticated supervised methods, simple unsupervised summarizers perform unexpectedly well. Many of them achieved the state-of-the-art performance when they were proposed (Erkan and Radev, 2004; Gillick et al., 2009) and still serve as competitive baselines (Hong et al., 2014).

After choosing the basic systems, we present a two-step framework of system combination. At the first step, we generate candidate summaries. We investigate two methods to do this: one uses entire basic summaries directly, the other combines these summaries on the sentence level. Experiments in Section 6.4 show that the upper bound of the second method is much higher than that of the first one. Thus, we choose it over the other.

At the second step, we propose a new supervised model that computes content quality scores over all candidate summaries. Our model relies on a rich set of features. These features are derived from different sources, including the original input, the basic summaries, and summary-input pairs from the New York Times (NYT) corpus. We also design features to capture the intuition that content from a better system have a higher chance to be selected. The summary with the highest score is chosen.

Our system is evaluated on nine DUC and TAC datasets. While generating short summaries (i.e., 50, 100 words), our system performs better than the best basic system in terms of automatic evaluation (comparable on ROUGE-2, better on ROUGE-1) and manual evaluation of the Pyramid score (evaluated on one dataset). It also performs on par with the state-of-the-art systems on multiple datasets. However, our model does not outperform the best basic system while generating longer summaries (i.e., 200, 250, 400 words). We show that two major problems need to be tackled as the summaries get longer: (1) it becomes intractable to generate all candidate summaries, (2) it becomes practically more difficult to outperform the best basic system, which we will show in Section 6.7.4.

This chapter is organized as follows. Section 6.2 reviews prior work. In Section 6.3, we describe the datasets used in this study. Section 6.4 describes two methods of generating candidate summaries and discusses the upper bound (oracle performance) of these two methods. Section 6.5 presents the features used in our supervised model. Section 6.6 introduces several baseline approaches. Experiments and results are shown in Section 6.7, followed by a conclusion.

6.2 Related Work

System combination has enjoyed great success on many natural language processing (NLP) domains, such as automatic speech recognition (ASR) (Fiscus, 1997; Mangu et al., 2000), machine translation (MT) (Frederking and Nirenburg, 1994; Bangalore et al., 2001; Rosti et al., 2007), part-of-speech tagging (Van Halteren et al., 1998; Tür and Oflazer, 1998) and parsing (Henderson and Brill, 1999; Sagae and Lavie, 2006; Zhang et al., 2009). However, only a handful of papers have utilized this idea for summarization. Mohamed and Rajasekaran (2005) present a method that relies on a document graph (DG), which includes concepts connected by relations. This method selects among the output of the basic systems, based on its overlap compared to the input in terms of DG. Thapar et al. (2006) propose a method that iteratively includes sentences, according to the overlap of DG between the current sentence and (1) the original input, or (2) the set of basic summaries. However, in both papers, the machine summaries are not compared against human summaries. Rather, their evaluations compare the summaries to the input based on the overlap of DG, which means no "real" evaluation is performed. Moreover, even when evaluated in this way, the combined system does not show advantages over the best basic system.

System combination in summarization has also been regarded as rank aggregation (Liu et al., 2007), where the combined system re-ranks the input sentences based on the ranks of these sentences assigned by the basic systems. Wang and Li (2012) employ an unsupervised method to minimize the distance of the final ranking compared to the initial rankings. Pei et al. (2012) propose a supervised method which handles an issue in Wang and Li (2012) that all basic systems are regarded as equally important. Even though both methods show advantages over the basic summarizers, they have two limitations. Most importantly, only summarizers that assign importance scores to each sentence can be used as the input summarizers. Second, only the sentence scores (ranks) from the basic systems and system identity information is utilized during the re-ranking process. Signals from the original input is ignored. Our method overcomes these limitations.

Our method derives an overall informativeness score for each candidate summary, then selects the one with the highest score. This is related to the growing body of research in global optimization, which selects the most informative subset of sentences towards a global objective (McDonald, 2007; Gillick et al., 2009; Aker et al., 2010). Some papers use Integer Linear Programming (ILP) to find the exact solution (Gillick et al., 2009; Li et al., 2015), others employ supervised approaches to (approximately) optimize the ROUGE scores of a summary (Lin and Bilmes, 2011; Kulesza and Taskar, 2012). We also use ROUGE scores while training our model.

In this chapter, we propose novel features to encode the content quality of a summary. Though prior work has extensively investigated features that are indicative of important words (Yih et al., 2007; Hong and Nenkova, 2014a) or sentences (Litvak et al., 2010; Ouyang et al., 2011), little work has focused on designing global features defined over the summary. Indeed, even for papers that employ supervised methods to conduct global inference (Aker et al., 2010; Kulesza and Taskar, 2012), the features are defined on the sentence level. The most closely related work studied automatic evaluation of summarization without human references (Louis and Nenkova, 2009; Saggion et al., 2010), where the effectiveness of several summary-input similarity metrics are examined. Another related paper defines paragraph-level features to determine whether the lead paragraph(s) of article is informative (Yang and Nenkova, 2014). In our work, we propose a wide range of features. These features are derived not only based on the input set, but also based on the basic summaries, as well as summary-input pairs of the New York Times corpus.

6.3 Data and Evaluation

We conduct a large scale experiment on nine datasets from the Document Understanding Conference (DUC) (2001–2007) and the Text Analysis Conference (TAC) (2008, 2009). The tasks include generic (2001–2004) and topic-based (2005–2009) multi-document summarization. The data is described in Section 2.1.

We mainly focus on generating 100 word summaries, which uses six datasets (DUC 2001–2004, TAC 2008, 2009). We also evaluate our model on generating summaries other than 100 words on five DUC datasets (2001, 2002, 2005–2007).

We use the ROUGE toolkit (Lin, 2004) for automatic evaluation. We report unigram (ROUGE-1) and bigram (ROUGE-2) recall, with stemming and not removing stopwords (see Section 2.2.2 for a detailed description of ROUGE).

6.4 Generating Candidate Summaries

We first introduce the four basic unsupervised systems, then describe our approach of generating candidate summaries. The four systems that we combine are ICSISumm (Gillick et al., 2008), Greedy-KL (Haghighi and Vanderwende, 2009), ProbSum (Nenkova and Vanderwende, 2005) and LLRSum (Conroy et al., 2006a). The code of ICSISumm[2] is made available by the authors, while we implement the other three systems. The four systems are described in Section 4.2.2. ICSISumm achieves the highest ROUGE-2 in TAC 2008, 2009 workshops and a state-of-the-art performance on the DUC 2004 dataset (see Table 4.5). Greedy-KL is a very strong baseline, which sometimes performs on par with more sophisticated systems. ProbSum and LLRSum are often used as baselines in recent work. ProbSum and LLRSum are implemented based on GreedySum (Section 4.1.2), where a sentence is considered as non-redundant if it is not similar to any sentences already in the summary, measured by cosine similarity on binary vector representations with stopwords included.

The four basic systems can be (or broadly) regarded as extractive summarization systems. Even though ICSISumm has a preprocessing component that removes obviously irrelevant content from the input sentences, this step is rather conservative.

The four systems are used for both generic and topic-based summarization. For the latter task, the "topic statement" is utilized as follows. For ICSISumm, the sentences that have no overlap with the "topic statement" in terms of content words are ignored when computing document frequency for bigrams (Gillick et al., 2008). For the three systems that we implemented, we exclude the sentences that have no overlap with the "topic statement" in terms of content words during generation; this step improves the performance for our systems.

Based on the output from the basic systems, we investigate two methods of generating candidate summaries. Here we focus on combining 100 word summaries. Section 6.4.3 compares the oracle performance (upper bound) of these two methods.

[2] https://code.google.com/p/icsisumm/

6.4.1 Selecting a Full Summary

There does not exist a system that always outperforms the others for all problems. Based on this fact, we directly select among the summary outputs from the basic systems. Methods that only use system output have been developed for system combination in machine translation (Nomoto, 2004; Paul et al., 2005).

6.4.2 Sentence Level Combination

Different systems provide different pieces of the correct answer. Based on this fact, the combined summary should include sentences that appear in summaries produced by different systems. The majority of system combination methods in NLP domains also reconstruct the final answer (Fiscus, 1997; Sagae and Lavie, 2006; Rosti et al., 2007) instead using the output directly. Here we generate candidate summaries by using a subset of sentences from the basic summaries. A similar approach has been used to generate candidate summaries for single-document summarization, where sentences from the document to be summarized are used (Ceylan et al., 2010).

Let $D = s_1, \ldots, s_n$ denote the sequence of unique sentences that appear in the basic summaries.[3] We enumerate all subsequences $A_i = s_{i_1}, \ldots, s_{i_k}$ of D in lexicographical order. A_i can be used as a candidate summary iff $\sum_{j=1}^{k} l(s_{i_j}) \geq L$ and $\sum_{j=1}^{k-1} l(s_{i_j}) < L$, where $l(s)$ is the number of words in s and L is the predefined summary length. Table 6.1 shows the average number of (unique) sentences and summaries that are generated per input.

Note here that we consider the order of sentences in A_i as a relatively unimportant factor. Though two summaries with the same set of sentences can have different ROUGE scores due to the truncation of the last sentence, however, because the majority of content covered are still the same, the difference in ROUGE score is relatively small. In order to generate all possible summaries that cover different

[3] We sort the sentences based on their lengths in descending order. Less number of candidate summaries will be generated in this way, which saves the storage space.

Dataset	# sents	# unique	# summaries	total summaries	Jaccard
DUC 2001	20.8	17.7	7498	224940	0.254
DUC 2002	21.1	17.6	12048	710832	0.265
DUC 2003	19.3	15.4	3448	103440	0.320
DUC 2004	19.5	15.6	3270	163500	0.321
TAC 2008	18.5	14.8	2436	107184	0.326
TAC 2009	18.0	13.7	1328	63744	0.340

Table 6.1: Average number of sentences (# sents), unique sentences (# unique) and candidate summaries per input (# summaries). We also show the total number of candidate summaries for each dataset and the average word-level Jaccard coefficient between the summaries from different systems.

content, one needs to swap the last sentence. However, as can be seen in Table 6.1, the total number of summaries per dataset is already huge. Therefore, we do not generate other candidate summaries, because it would cost much more additional time and space and the difference in ROUGE score is relatively small.

In Chapter 5, we have shown that the state-of-the-art systems generate summaries that have low overlap with each other. Here we also compute the overlap of word types between summaries from different systems in terms of Jaccard coefficient, with stemming[4] and stopwords included. The system level Jaccard coefficient is equal to the mean of summary level Jaccard coefficients. As shown in Table 6.1, the word overlap between the summaries are still very low, which indicates the systems that we have combined have high diversity.

6.4.3 Oracle Comparison

We examine the upper bounds of the two methods described in Section 6.4.1 and Section 6.4.2. For the first method, we design two oracle systems that pick the basic summary with the highest ROUGE-1 (R-1) and ROUGE-2 (R-2) (denoted as

[4] Here we use the Porter Stemmer (Porter, 1980)

	DUC 2001		DUC 2002		DUC 2003	
	R-1	R-2	R-1	R-2	R-1	R-2
ICSISumm	0.342	0.079	0.373	0.095	0.381	0.103
Greedy-KL	0.331	0.067	0.358	0.075	0.383	0.086
ProbSum	0.303	0.056	0.326	0.071	0.360	0.088
LLRSum	0.318	0.067	0.329	0.068	0.354	0.085
SumOracle R-1	**0.361**	0.084	**0.391**	0.103	**0.407**	0.106
SumOracle R-2	0.349	**0.090**	0.385	**0.106**	0.398	**0.113**
SentOracle R-1	**0.400**	0.097	**0.439**	0.121	**0.442**	0.123
SentOracle R-2	0.368	**0.109**	0.416	**0.134**	0.422	**0.136**

	DUC 2004		TAC 2008		TAC 2009	
	R-1	R-2	R-1	R-2	R-1	R-2
ICSISumm	0.384	0.098	0.388	0.119	0.393	0.121
Greedy-KL	0.383	0.090	0.372	0.094	0.384	0.099
ProbSum	0.354	0.082	0.350	0.087	0.357	0.094
LLRSum	0.359	0.081	0.372	0.096	0.364	0.097
SumOracle R-1	**0.403**	0.103	**0.408**	0.124	**0.417**	0.130
SumOracle R-2	0.394	**0.108**	0.403	**0.129**	0.411	**0.136**
SentOracle R-1	**0.437**	0.119	**0.448**	0.139	**0.453**	0.146
SentOracle R-2	0.420	**0.131**	0.430	**0.152**	0.437	**0.158**

Table 6.2: Performance of the basic and oracle systems based on the two methods described in Section 6.4.1 and Section 6.4.2. The ROUGE metric that each oracle optimizes are shown in **bold**.

SumOracle R-1 and SumOracle R-2). For the second method, we design two oracle systems that pick the best summary in terms of R-1 and R-2 among the summary candidates (denoted as SentOracle R-1 and SentOracle R-2). As shown in Table 6.2, the advantage of the first two oracles over ICSISumm is limited: on average 0.021/0.006 and 0.013/0.011 (R-1/R-2). However, the advantage of the latter oracles over ICSISumm is much larger: on average 0.060/0.022 and 0.039/0.034 (R-1/R-2). Clearly, it is more promising to combine the basic summaries at the sentence level. Therefore, we adopt the latter method to generate candidate summaries.

6.5 Features

We introduce the features used in our model that selects among the candidate summaries. Traditionally in summarization, features are derived based on the input (denoted as I). In our work, we propose a class of novel features that scores the content quality by comparing the summary to the set of basic summaries (denoted as H), where H can be regarded as a *hyper-summary* of I. This excels in the way that it takes advantage of consensus between systems. Moreover, we propose system identity features, which capture the fact that content from a better system should have a higher chance to be selected.

Our model includes classical indicators of content importance (e.g., frequency, locations) and features that have been recently proposed for other tasks. For example, we use features that estimate the intrinsic importance of words from a large corpus (Chapter 5). We also include features that compute the information density (Yang and Nenkova, 2014) of the first sentence that each word appears in. We tailor these two classes of features specifically for system combination (see Section 6.5.2).

We classify our features into summary level, word level and system identity features. There are 360 features in our model. We do not consider stopwords and do not perform stemming.

6.5.1 Summary Level Features

Summary level features encode the informativeness of the entire summary directly. Some of them are initially proposed in Louis and Nenkova (2013) that evaluates the summary content without human models. Different from them, the features in our work use either I or H as the "*input*" (except for the redundancy features). "*Input*" refers to I or H in the rest of Section 6.5. This class includes 26 features.

Distributional Similarity: These features compute the distributional similarity (divergence) between the n-gram ($n = 1, 2$) probability distribution of the summary and that of the *input* (I or H). Good summaries tend to have high similarity and low divergence. Three similarity measures are used here: Kullback-Leibler (KL) divergence, Jenson-Shannon (JS) divergence and cosine similarity.

Let P and Q denote the n-gram distribution of the summary and that of the input respectively. Let $p_\lambda(w)$ be the probability of n-gram w in distribution λ. The KL divergence $KL(P \parallel Q)$ and the JS divergence $JS(P \parallel Q)$ are defined as:

$$KL(P \parallel Q) = \sum_{w} p_P(w) \cdot \log \frac{p_P(w)}{p_Q(w)} \tag{6.1}$$

$$JS(P \parallel Q) = \frac{1}{2} \cdot KL(P \parallel A) + \frac{1}{2} \cdot KL(Q \parallel A) \tag{6.2}$$

where A is the average of P and Q. Noticing that KL divergence is not symmetric, both $KL(P \parallel Q)$ and $KL(Q \parallel P)$ are computed. In particular, smoothing is performed while computing $KL(Q \parallel P)$.[5] Smoothing is not performed while computing $KL(P \parallel Q)$, since all sentences that appeared in the machine summaries are extracted from the input.

Topic words: Good summaries tend to include more topic words (TWs) (i.e., words that are descriptive of the *input*). We derive TWs using the log-likelihood ratio (LLR) test, which is described in Section 3.4.2. For each summary, we compute: (1) the ratio of the words that are TWs to all words in the summary; (2) the recall of TWs in the summary.

Sentence location: Sentences that appear at the beginning of an article are likely to be more critical. Greedy-based summarizers (ProbSum, LLRSum, GreedyKL) also select important sentences first. To capture these, we set features over the sentences in a summary (S) based on their locations. There are features that indicate whether a sentence in S has appeared as the first sentence in the *input* (I or H). We also set features to indicate the normalized position of a sentence within documents of an *input*: by assigning 1 to the first sentence, 0 to the last sentence. When one

[5] This is necessary, because $KL(Q \parallel P)$ is undefined when $p_P(w) = 0$ and $p_Q(w) > 0$. We use the same setting for smoothing as in Louis and Nenkova (2013).

sentence appears multiple times, the earliest position is used. Features are then set on the summary level, which equal to the mean of their corresponding features on the sentence level over all sentences in the summary.

Redundancy: Redundancy correlates negatively with content quality (Pitler et al., 2010). To indicate redundancy, we compute the maximum and average cosine similarity of all pairs of sentences in the summaries. Summaries with higher redundancy are expected to score higher on these two features.

6.5.2 Word Level Features

Better summaries should include words or phrases that are of higher importance. Hence, we design features to encode the overall importance of unigrams and bigrams in a summary. We first generate features for the n-grams ($n = 1, 2$) in a summary S, then derive the final feature vector $\mathbf{v_S}$ for S. The procedure is as follows:

Let t denote the unigram or bigram in a summary. For each t that includes content words, we form $\mathbf{v_t}$, where each component of $\mathbf{v_t}$ is an importance indicator of t. If t does not include any content words, we set $\mathbf{v_t} = \mathbf{0}$, which means t is uninformative. Let S' denote the unique n-grams in S and let L denote the summary length, we compute two feature vectors defined over S: $\mathbf{v_{S_1}} = (\sum_{t \in S} \mathbf{v_t})/L$ and $\mathbf{v_{S_2}} = (\sum_{t \in S'} \mathbf{v_t})/L$. $\mathbf{v_{S_1}}$ and $\mathbf{v_{S_2}}$ are the coverage of n-grams by word token and word type, which represents two extreme ways of handling repetitive words. Finally, $\mathbf{v_S}$ is formed by concatenating $\mathbf{v_{S_1}}$ and $\mathbf{v_{S_2}}$ for unigrams and bigrams.

Below we describe the features in $\mathbf{v_t}$. The features are computed from the input (I), the hyper-summaries (H) and a large corpus of summary-input pairs.

Frequency related features: For each n-gram t, we compute its probability, TF*IDF[6] and document frequency (DF). If t is a unigram, we include the χ-square statistic from LLR test ($LLR(t)$). If t is a bigram $t = t_1t_2$, we use $LLR(t_1)+LLR(t_2)$

[6] Inverse document frequency is derived using the articles between year 2004 and 2007 from the New York Times corpus. We perform add one smoothing to handle the out of vocabulary words.

instead of $LLR(t)$.[7] Another feature is set to be equal to DF normalized by the number of input documents. This is because in some datasets (e.g., DUC 2001, 2002), the number of documents in the input is not a constant number. A binary feature is set to determine whether DF is at least three, inspired by the finding that document specific words should not be regarded as informative (Mason and Charniak, 2011).[8]

It has been shown that unimportant words (e.g., low frequency words) of an input should not be considered while scoring the summary (Gupta et al., 2007; Mason and Charniak, 2011). The features below are designed to capture this, where we approximate the important n-grams as the ones that include topic words. Let the binary function $b(t)$ denote whether or not t includes topic words. Features are set to be equal to the product of the DF related features and $b(t)$.

Word locations: Words that appear close to the beginning of I or H are likely to be important. For each n-gram token in the *input*, we compute its normalized position in the document. Then for each n-gram type t, we compute its *first*, *average*, *last* and *average first* location across its occurrences in all documents. Features are also set to determine whether t has appeared in the first sentence and the total number of times that it appears in the first sentences of the *input*.

Information density of the first sentence: The first sentence of an article can be either informative or entertaining. Clearly, words that appear in an informative first sentence should be assigned higher importance scores. To capture this, we compute the importance score (called information density in Yang and Nenkova (2014)) of the first sentence, that is defined as the number of TWs divided by the number of words in the sentence. For each t, we compute the maximum and average of importance scores over all first sentences that t appears in.

Intrinsic word importance (Global indicators): Some words are intrinsically

[7] We compute in this way, because the bigrams in the background corpus are sparse.

[8] ICSISumm optimizes the coverage of key bigrams, where the bigrams that appear in at least three human summaries are considered. Hence, we choose three in our setting as well.

important (e.g., war, death) or unimportant (e.g., Mr., a.m.) to humans, regardless of the input. Here we estimate the intrinsic importance of words, using 160K summary-article pairs from the New York Times corpus (Sandhaus, 2008). Specifically, five scores are used to denote the importance of each word (Formula 5.1 to Formula 5.5 in Chapter 5), which are derived by Method 2 described in Section 5.2.

Let L denote a list of words ranked by one of the methods that measures the intrinsic importance. For each word in S, features are set to determine whether or not the word is ranked within the top-k and bottom-k in L (one feature for top-k, one for bottom-k), which indicates whether people consider the word to be among the top k most intrinsically important (unimportant) ones.[9] This class of features are set for only unigrams.

6.5.3 System Identity Features

System identity features capture the intuition that words or sentences from a better system are more likely to appear in a good combined summary. For each basic system A_i, we compute the sentence and n-gram overlap between S and the summary from A_i (S_{A_i}). We hypothesize that the quality (i.e., ROUGE score) of a summary is positively (negatively) correlated with the overlap between this summary and a good (bad) basic summary of the same input. We design six sentence and two word overlap features for each individual system, which leads to a total of 32 features.

Sentence overlap: Let D_0, D_{A_i} denote the set of sentences in S and S_{A_i}, respectively. For each system A_i, we set a feature $|D_0 \bigcap D_{A_i}|/|D_0|$. We further consider sentence lengths: let $l(D)$ denote the total length of sentences in set D, a feature $l(D_0 \bigcap D_{A_i})/l(D_0)$ is set for each system A_i. Lastly, we compute the binary version of $|D_0 \bigcap D_{A_i}|/|D_0|$.

Furthermore, we exclude sentences that appear in multiple basic summaries from D_0 and compute the three features above. System identity features might be more

[9] The values are: 100, 200, 500, 1000, 2000, 5000, 10000.

helpful in selecting among the sentences that are generated by only one of the systems.

N-gram overlap: Let t denote an n-gram (n=1,2) type and let $Count_S(t)$ denote the number of times t appeared in S. For each system A_i, we set a feature:

$$\frac{\sum_{t \in S} min(Count_S(t), Count_{S_{A_i}}(t))}{\sum_{t \in S} Count_S(t)} \tag{6.3}$$

which indicates the fraction of n-gram tokens in S that appears in S_{A_i}. The n-grams consisting of solely stopwords are removed before computation.

6.6 Baseline Approaches

We present three summary combination methods as baseline methods. Table 6.3 compares the baseline methods and our proposed method (SumCombine).

Voting based summary generation: The idea of voting has been used for automatic speech recognition, where a model selects the best output word sequence within a word transition network by "rescoring" or "voting" (Fiscus, 1997). It has also been used in parsing, where a model decides whether a constituent should be used in a parse by majority voting (Henderson and Brill, 1999). Here we select sentences according to the total number of times that they appear in all basic summaries, from large to small. When there are ties, we randomly pick an unselected sentence. The procedure is repeated 100 times and the mean ROUGE score is reported.

Summarization from summaries (Sum): We directly run summarizers over the summaries from the basic systems. We report the performance of ICSISumm and Greedy-KL, as they perform better than the other two summarizers.

Jensen-Shannon (JS) divergence: This method selects among the pool of candidate summaries. The summary with the smallest JS divergence between the summary and the input (JS-I) or the hyper-summaries (JS-H) is selected. Summary-input JS divergence is the best metric to identify better summarizers without human references (Louis and Nenkova, 2009). This baseline is also similar to the methods proposed in Thapar et al. (2006), where a graph-based similarity measure is used.

	Voting	Sum	JS-H	JS-I	SumCombine
Supervised?	No	No	No	No	Yes
Need candidate summaries?	No	No	Yes	Yes	Yes
Signal from the input?	No	No	No	Yes	Yes
Signal from the basic summaries?	Yes	Yes	Yes	No	Yes
Signal from the NYT corpus?	No	No	No	No	Yes

Table 6.3: Comparison between different combination methods in terms of strategies.

6.7 Experiments and Results

6.7.1 Experimental Settings

We use the DUC 03, 04 datasets as training and development sets. The candidate summaries of these two sets are used as training instances. There are 80 input sets; each input includes an average of 3336 candidate summaries. During development, we perform four-fold cross-validation. The DUC 01, 02 and TAC 08, 09 datasets are used as the held-out test sets. In order to find a better learning method, we experiment with a regression model and a learning-to-rank model, which we will describe next.

The first model that we use is support vector regression (SVR) (Drucker et al., 1997). Let $\{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2) \ldots (\mathbf{x_n}, y_n)\}$ denote the training data, where $\mathbf{x_i}$ denotes the feature space of the input samples (i.e., all candidate summaries) and y_i denotes the ROUGE scores of the summaries. The goal is to find a function $f(\mathbf{x}) = \mathbf{x} \cdot \mathbf{w} + b$

that has at most ϵ deviation from y_i in the training data. This can be represented as a convex optimizing problem, which aims to minimize $\frac{1}{2}||\mathbf{w}||^2$, subjective to the constraints of $|f(\mathbf{x_i})-y_i| \leq \epsilon$. Since the function $f(\mathbf{x})$ may not be always feasible, one needs to introduce slack variables for the optimization problem, which is the same as the soft margin technique in support vector machine (SVM) (Cortes and Vapnik, 1995). The slack variables also avoids overfitting. In our work, we use the SVR model implemented in SVMLight (Joachims, 1999). While prediction, the output of SVR are real numbers, which approximate the ROUGE scores of the summaries. SVR has been previously used for estimating sentence (Ouyang et al., 2011) or document (Aker et al., 2010) importance in summarization.

The second model that we use is SVM-Rank (Joachims, 2002), a learning to rank algorithm that only considers the relative rank between training samples. A natural application of SVM-Rank is to rank web documents towards a query. Clearly, our problem of selecting the best summary among candidates is similar to that, as the input can be regarded as the queries and the candidate summaries can be regarded as the web documents. Specifically, SVM-Rank aims to minimize the total number of discordant pairs (a discordant pair is a pair of summaries that are ranked in the wrong order) within the training data. Here we use the SVM-Rank toolkit (Joachims, 2006). While prediction, the toolkit outputs real numbers, which are used to decide the relative ranks between summaries of the same input. SVM-Rank has been employed for ranking summaries according to their linguistic qualities in prior work (Pitler et al., 2010).

We experiment with two ways of assigning labels towards the training instances: ROUGE-1 (R-1) and ROUGE-2 (R-2). This results in four different combinations of models and labels. While training on R-1 using SVR, we use linear kernel[10] with default parameters. While training on R-2 using SVR, the default ϵ (0.1) in loss

[10] Apart from linear kernel, we have experimented with polynomial kernel and RBF kernel. However, neither outperforms the linear kernel. Thus we use linear kernel, which is also the simplest and fastest setting.

Settings	ROUGE-1	ROUGE-2
SVR + ROUGE-1	0.3986	0.1040
SVR + ROUGE-2	0.3890	0.1023
SVMRank + ROUGE-1	0.3932	0.0996
SVMRank + ROUGE-2	0.3854	0.0982

Table 6.4: Performance on the development set (DUC 2003, 2004 data) by four fold cross-validation.

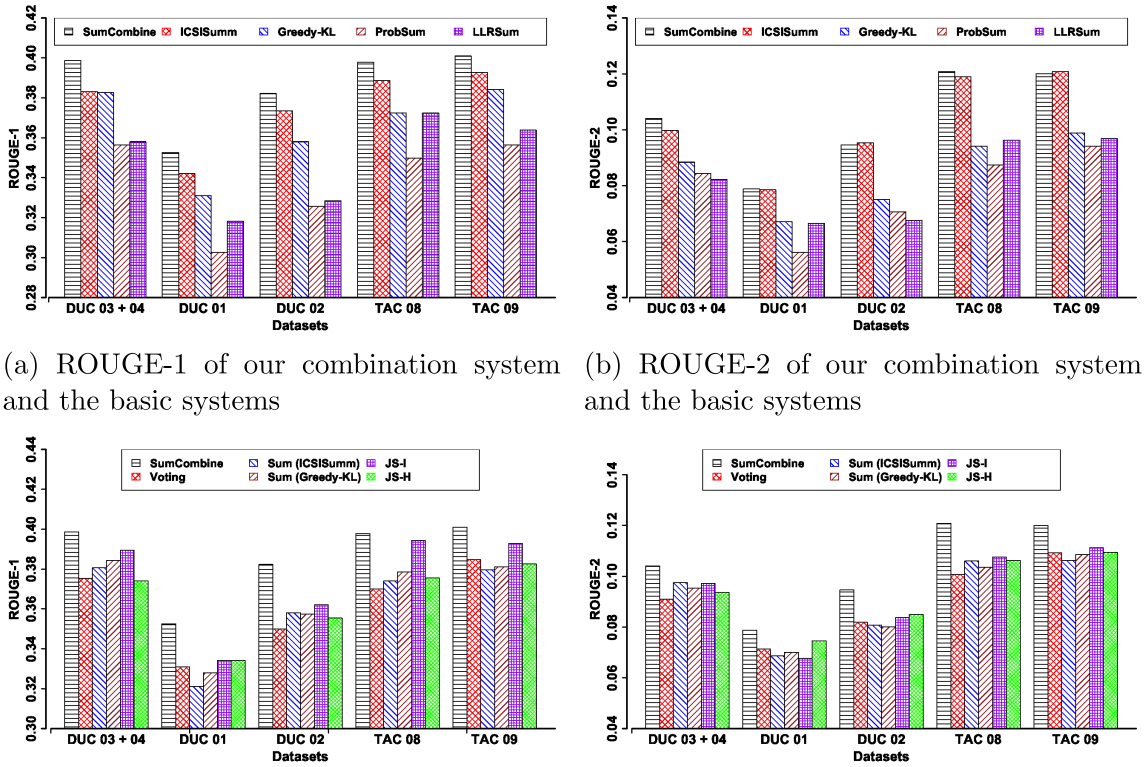
function is too large, because R-2 ranges between 0 and 0.2. This causes $\mathbf{w} = 0$ and all the summaries assigned to the value b. Therefore, we have tuned ϵ, with possible values range between 0.01 and 0.09 with a step of 0.01. $\epsilon = 0.03$ is the best setting. For SVM-Rank, we use linear kernel and default parameters.

Table 6.4 shows the performance of these four combinations. SVR outperforms SVM-Rank, which means that it is useful to compare the summaries across different input sets and make use of the actual ROUGE scores while training.

We choose ROUGE-1 (R-1) as the metric for training, because it outperforms ROUGE-2 (R-2) for both models. Especially for SVR, even though we have to tune ϵ while training on R-2, the best performance never exceeds that of R-1, where the default parameters are used. We suspect that the advantage of R-1 is because it has higher sensitivity (recall) in capturing the differences in content between summaries. Indeed, recent methods that optimized content quality on the summary level mostly use R-1 while training (Lin and Bilmes, 2011; Kulesza and Taskar, 2012; Sipos et al., 2012). They may have observed a similar phenomenon.

6.7.2 Combining 100 Word Summaries

We evaluate our model (SumCombine) on generating 100 word summaries. We report the performance on the development sets (DUC 2003, 2004) and the held-out test sets (DUC 2001, 2002 and TAC 2008, 2009).



(a) ROUGE-1 of our combination system and the basic systems

(b) ROUGE-2 of our combination system and the basic systems

(c) ROUGE-1 of our combination system and the baseline approaches

(d) ROUGE-2 of our combination system and the baseline approaches

Figure 6.1: ROUGE-1 and ROUGE-2 for the task of generating 100 word summaries on the DUC 2001–2004 and TAC 2008, 2009 datasets. We compare our system to the basic systems in (a), (b) and the baseline approaches in (c), (d).

Comparing with the Basic Systems and the Baseline Methods

First, we compare SumCombine to the basic systems. As shown in Figure 6.1 (a), SumCombine performs consistently better than all basic systems on ROUGE-1 (R-1). When comparing SumCombine to the best basic system (ICSISumm), the difference is significant on the DUC 2002, 2003, and 2004 data and close to significant on the DUC 2001 and TAC 2009 data (Table 6.5). However, our model performs on par with ICSISumm on ROUGE-2 (R-2), where we only observe a significant improvement on the DUC 2004 dataset. Our model performs significantly better than the other basic systems on both R-1 and R-2 (Figure 6.1 (a), (b)).

We manually evaluate the summaries using the Pyramid method (Nenkova et

	DUC 2001		DUC 2002		DUC 2003	
	R-1	R-2	R-1	R-2	R-1	R-2
ICSISumm	0.3421	0.0785	0.3735	0.0953	0.3813	0.1028
SumCombine	0.3526†	0.0788	**0.3823**	0.0946	**0.3959**	0.1018

	DUC 2004		TAC 2008		TAC 2009	
	R-1	R-2	R-1	R-2	R-1	R-2
ICSISumm	0.3841	0.0978	0.3886	0.1190	0.3926	0.1208
SumCombine	**0.3995**	**0.1048**	0.3978	0.1208	0.4009†	0.1200

Table 6.5: Performance comparison between ICSISumm and our method. **Bold** and † represent statistical significant ($p < 0.05$) and close to significant ($0.05 \leq p < 0.1$) compared to ICSISumm (two-sided Wilcoxon signed-rank test).

	Oracle	SumCombine	ICSISumm	Greedy-KL
The Pyramid score	0.626	0.549	0.530	0.459

Table 6.6: The Pyramid scores on the TAC 2008 dataset.

al., 2007) on the TAC 2008 data (see Section 2.2 for a detailed description of the Pyramid method). Since the Pyramid method is time consuming, we only evaluate on the TAC 2008 data. As shown in Table 6.6, our model outperforms ICSISumm and Greedy-KL by 0.019 and 0.090, respectively. However, it falls behind the oracle system by 0.078. The advantage of SumCombine over ICSISumm is not significant, while the advantage over Greedy-KL is significant. In fact, if we look into the p-value derived by comparing SumCombine to ICSISumm on R-1, TAC 2008 is the least significant dataset. Hence, the advantage of SumCombine over ICSISumm on the other datasets is likely to be larger in terms of manual evaluation.

Figure 6.1 (c), (d) compare our model with the baseline methods proposed in Section 6.6. The baselines that only consider consensus between different systems perform poorly (summarization on summaries, voting, JS-H). Indeed, the systems that we have combined are of different performances. If we only use consensus between systems as the single indicator, the final performance is likely to be dragged down by low-performing systems. JS-I selects the candidate summary with the smallest input-summary JS divergence, which achieves the best ROUGE-1 among baselines. However, it is still much inferior to our model. Nevertheless, we will show in Section 7.2 that JS-I has very competitive performance, which only falls behind that of SumCombine[11] by a small margin, based on 24 different combinations of the input systems on the TAC 2008 data.

Comparing with the State-of-the-art Systems

Table 6.7 compares our model with the state-of-the-art systems. On the DUC 2003 and 2004 data, ICSISumm is among one of the best systems. Our model performs significantly better compared to it on R-1. We also achieve a better performance compared to the other top performing extractive systems (DPP, RegSum) on the DUC 2004 data (these systems are introduced in Section 4.2).

On the DUC 2001 and 2002 data, the top performing systems we find are R2N2_ILP (Cao et al., 2015a) and PriorSum (Cao et al., 2015b); both of them leverage neural networks.[12] Compare to these two, SumCombine achieves a lower performance on the DUC 2001 data and a higher performance on the DUC 2002 data. It also has a slightly lower R-1 and higher R-2 compared to ClusterCMRW (Wan and Yang, 2008), a graph-based system that achieves the highest R-1 on the DUC 2002 data. On the TAC 2008 dataset, the best performing systems (Li et al., 2013a; Almeida and Martins, 2013) achieve the state-of-the-art performance by sentence compression. Our model performs extractive summarization, but still has

[11] However, we do not use the system identity feature there. Not including this feature class will slightly decrease the performance, as we will show in Section 6.7.3.

[12] These two systems truncate the summaries to 665 words on the DUC 2004 dataset. Thus we do not compare with them on the DUC 04 data.

Dataset	System	ROUGE-1	ROUGE-2
DUC 2003	ICSISumm	0.3813	0.1028
	SumCombine	0.3959	0.1018
DUC 2004	ICSISumm	0.3841	0.0978
	SumCombine	0.3995	0.1048
	DPP	0.3979	0.0962
	RegSum	0.3857	0.0975
DUC 2001	ICSISumm	0.3421	0.0785
	SumCombine	0.3526	0.0788
	R2N2_ILP	0.3691	0.0787
	PriorSum	0.3598	0.0789
DUC 2002	ICSISumm	0.3733	0.0954
	SumCombine	0.3823	0.0946
	R2N2_ILP	0.3796	0.0888
	PriorSum	0.3663	0.0897
	ClusterCMRW	0.3855	0.0865
TAC 2008	ICSISumm	0.3880	0.1186
	SumCombine	0.3978	0.1208
	Li et al. (2013a)	n/a	0.1235
	Almeida & Martins (2013)	n/a	0.1230
	Li et al. (2015)	n/a	0.1184
TAC 2009	ICSISumm	0.3931	0.1211
	SumCombine	0.4009	0.1200
	Li et al. (2015)	n/a	0.1277

Table 6.7: Performance comparison on six DUC and TAC datasets.

similar R-2 compared to theirs.[13] On the TAC 2009 data, the best system uses a supervised method that weighs bigrams in the ILP framework by leveraging external resources (Li et al., 2015). This system is better than ours on the TAC 2009 data and is inferior to ours on the TAC 2008 data.

Overall, our combination model achieves very competitive performance, which is comparable to the state-of-the-art on multiple benchmarks.

[13] These two work report ROUGE-SU4 (R-SU4) (measures skip bigram with a maximum gap of 4) instead of R-1. Our model achieved similar R-SU4 (−0.0002/+0.0007) compared to them.

Comparing with Other Combination Methods

At last, we compare our system to SSA (Pei et al., 2012) and WCS (Wang and Li, 2010; Wang and Li, 2012). These models perform system combination by rank aggregation and are evaluated on the DUC 04 data. In order to be consistent with these two papers, we truncate our summaries to 665 bytes and report F_1-score. In order to compare with them, we also report ROUGE-SU4 (R-SU4), which measures the skip bigram with a maximum gap of 4 words. Pei et al. (2012) report the performance on 10 randomly selected input sets. In order to have the same size of training data, we conduct five-fold cross-validation. Wang et al. (2012) report the performance on all input sets of the DUC 2004 data.

As can be seen in Table 6.8, SumCombine performs better than SSA and WCS on R-2 and R-SU4, but not on R-1. It is worth noting that these three systems cannot be directly compared, because different basic systems are used. In fact, compared to SumCombine, SSA and WCS have larger improvement over the basic systems that are respectively used. This might be because rank aggregation is a better strategy, because combining weaker systems is easier to result in large improvements (we will show evidence of this in Section 7.2.4), or because of both reasons. It is inconclusive to draw a conclusion on which system is better.

System	ROUGE-1	ROUGE-2	ROUGE-SU4
SumCombine	0.3943	0.1015	0.1411
SSA (Pei et al., 2012)	0.3977	0.0953	0.1394
WCS (Wang and Li, 2012)	0.3987	0.0961	0.1353

Table 6.8: Comparison with other combination methods on the DUC 2004 dataset. SSA only report the performance on 10 input sets.

It is worth noting that SSA does not significantly outperform two of the four basic systems that they used on any metrics. The main reason is because they only use 10 sets for evaluations. Therefore, it is unclear how well SSA can perform on the full DUC 2004 dataset. WCS has demonstrated significant advantages over the basic systems that they combined. However, one parameter used in WCS have considerable effects to the final performance, which is also decided on the DUC 2004 data (Wang and Li, 2010). Though, the authors later show that their method is successful on the DUC 2002 data (Wang and Li, 2012).

6.7.3 Effects of Features

We conduct two experiments to examine the effectiveness of features (see Table 6.9). First, we remove one class of feature at a time from the full feature set. Second, we show the performance of a single feature class. Apart from reporting the performance on the development and the test sets, we also show the macro average performance across the five sets.[14] This helps us to understand the contribution of different features in general.

Summary level, word level and system identity features are all useful, with ablating them leads to an average of 0.0031 to 0.0041 decrease on R-1, 0.024 to 0.037 decrease on R-2. Ablating summary and word level features can lead to a significant decrease in performance on some datasets. If we use a single set of features, then summary and word level features are more useful than system identity features.

The word and summary level features compute the content importance based on three sources: the input, the basic summaries (hyper-summary) and the New York Times corpus (global). We ablate the features derived from these three sources respectively. Input-based features are the most important, with removing them leads to a very large decrease in performance, especially on R-1. Features derived from the basic summaries are also effective (hyper-summary): though the decrease in performance is small, we can observe a decrease on all datasets. Only using input based features is better than only using hyper-summary based features. The decrease

[14] We do not compute the statistical significance for the average score.

ROUGE-1	Dev. Set	DUC 01	DUC 02	TAC 08	TAC 09	Avg.	Diff
All features	0.3986	0.3526	0.3823	0.3978	0.4009	0.3864	NA
-summary	0.3946	0.3469	**0.3760**	0.3950	0.3988	0.3823	-0.0041
-word	**0.3946**	**0.3429**	0.3787	0.3939	0.4046	0.3829	-0.0035
-system	0.3964	0.3483	0.3772	0.4009	0.3936	0.3833	-0.0031
-input	**0.3822**	**0.3433**	0.3786	**0.3858**	0.3960	0.3772	-0.0092
-hyper-summary	0.3978	0.3512	0.3806	0.3968	0.3994	0.3852	-0.0012
-hyper & system	0.3983	0.3450	0.3778	0.4015	0.4059	0.3857	-0.0007
-global	0.3948	0.3457	0.3821	0.3959	0.4010	0.3839	-0.0025

ROUGE-2	Dev. Set	DUC 01	DUC 02	TAC 08	TAC 09	Avg.	Diff
All features	0.1040	0.0788	0.0946	0.1208	0.1200	0.1036	NA
-summary	0.1014	0.0779	0.0872	0.1185	0.1191	0.1008	-0.0028
-word	0.1002†	**0.0733**	0.0919	0.1172	0.1232	0.1012	-0.0024
-system	0.1022	0.0776	0.0895	0.1193	**0.1110**	0.0999	-0.0037
-input	**0.0956**	0.0764	0.0912	0.1148	0.1159	0.0988	-0.0048
-hyper-summary	0.1022	0.0777	0.0918	0.1193	0.1177	0.1017	-0.0019
-hyper & system	0.1014	0.0743	0.0918	0.1181	0.1186	0.1008	-0.0028
-global	0.1021	0.0760	0.0954	0.1136	0.1215	0.1017	-0.0019

ROUGE-1	Dev. Set	DUC 01	DUC 02	TAC 08	TAC 09	Avg.	Diff
All features	0.3986	0.3526	0.3823	0.3978	0.4009	0.3864	NA
summary	0.3960	**0.3344**	0.3748	0.3957	0.4009	0.3804	-0.0060
word	**0.3919**	0.3492	0.3784	0.3956	0.3956	0.3821	-0.0043
system	**0.3881**	**0.3430**	**0.3689**	0.3898	0.3926	0.3765	-0.0099
input	0.3979	**0.3410**	0.3764	0.3907	0.4015	0.3815	-0.0049
hyper-summary	**0.3852**	0.3447	**0.3665**	0.3871†	0.3906†	0.3748	-0.0116

ROUGE-2	Dev. Set	DUC 01	DUC 02	TAC 08	TAC 09	Avg.	Diff
All features	0.1040	0.0788	0.0946	0.1208	0.1200	0.1036	NA
summary	0.1018	**0.0701**	0.0910	0.1166	0.1170	0.0993	-0.0043
word	0.1006	0.0765	0.0905	0.1166	0.1146	0.0998	-0.0038
system	**0.0958**	0.0746	**0.0868**	**0.1096**	0.1145	0.0963	-0.0073
input	0.1009	0.0729†	0.0904	0.1129	0.1189	0.0992	-0.0044
hyper-summary	**0.0952**	**0.0725**	**0.0823**	**0.1080**	0.1140	0.0944	-0.0092

Table 6.9: Performance after ablating features (top two tables) or using a single class of features (bottom two tables). **Bold** and † represent statistical significant ($p < 0.05$) and close to significant ($0.05 \leq p < 0.1$) compared to using all features (two-sided Wilcoxon signed-rank test). Diff is the difference in performance compared to using all features.

of ablating global indicators is on average 0.002 for both R-1 and R-2. Global indicators are fairly effective on the development set and the DUC 01 dataset.

Interestingly, ablating hyper-summary and system identity features perform better than ablating either of the two, which sometimes outperforms using full features on R-1. Here we give an explanation. Hyper-summary provides indicators based on consensus between summaries. Because all systems are treated as equally important, this might give too much focus on low-performing systems. System identity features consider which system is better, which might give too much focus on top-performing systems. Hence, it makes sense that consider none or both feature classes is better than considering one of them. Using the full feature set is still the best choice overall.

WCS and SSA achieve improvements based on the output from basic systems. However, making predictions based on hyper-summaries performs poorly in our experiments (Table 6.9). This can be due to three reasons. First, the systems that they combine have similar performances, which makes voting more effective. Second, the systems that they combine are of lower performance, which are easier to be improved. Third, these systems use sentence outputs from different systems, which might include other effective signals compared to using the output summaries.

The effectiveness of the same feature class varies to a great extent across different datasets. For example, ablating word level features decreases R-2 significantly on the DUC 01 data, but increases R-2 on the TAC 09 data. However, by looking at the average performance, it becomes clear that it is necessary to use all features. Features computed based on the input are identified as the most important.

6.7.4 Combining Shorter or Longer Summaries

We examine our model on combining summaries other than 100 words. The summaries are evaluated by comparing with the human summaries of the same length, which are made available by the DUC/TAC workshops (see Table 2.1). SumCombine is trained on the DUC 2003 and 2004 data.

First, we combine 50 word summaries, which are evaluated on the DUC 2001 and 2002 datasets. Because the basic and the combined summaries are shorter, the number of candidate summaries is smaller. As shown in Table 6.10, our model performs better than all basic systems in terms of ROUGE-1 (R-1) and at least as good as the best basic system (ICSISumm) in terms of ROUGE-2 (R-2).

Second, we combine summaries longer than 100 words. The following datasets are used: the DUC 2001 data (200, 400 words), the DUC 2002 data (200 words) and the DUC 2005–2007 data (250 words). A natural difficulty arises when combining long summaries: it is intractable to enumerate all candidates. Hence, we produce 10000 unique candidate summaries for each input by randomly selecting sentences from the basic summaries. As shown in Table 6.10, 6.11, the performances of the two oracle systems that pick the candidate summaries with the highest R-1 and R-2 are much higher than that of the basic systems. Therefore, the potential is still high.

Though the potential is high, our model cannot generate summaries better than ICSISumm. As shown in Table 6.10 and Table 6.11, SumCombine achieves similar R-1 and inferior R-2 while generating 200 or 250 word summaries and achieves inferior R-1 and R-2 while generating 400 word summaries.

During our early experiments, we have experimented with two other methods of generating candidate summaries. Let N denote the number of summaries that we want to generate per input. The first method selects sentences based on the number of times that they appear in basic summaries. When there are ties, we randomly pick an unselected sentence. We repeat this procedure until N candidates are generated or all candidates are enumerated. The second method employs a beam search algorithm. We use breath-first search to build the search tree. At each level of the search tree, we iteratively append a potential sentence. The summaries at each level are sorted in increasing order of the heuristic cost, which is defined to be the Jensen-Shannon (JS) divergence between the current summary and the input. If a summary has reached the length limit (i.e., cannot be expanded), we append it to

Length	50		100		200		400	
	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
ICSISumm	0.279	0.064	0.342	0.079	0.419	0.120	0.496	0.155
Greedy-KL	0.262	0.046	0.331	0.067	0.400	0.092	0.480	0.129
ProbSum	0.249	0.040	0.304	0.055	0.368	0.077	0.441	0.112
LLRSum	0.250	0.046	0.313	0.065	0.385	0.093	0.453	0.121
SumCombine	0.286	0.064	0.353†	0.079	0.423	**0.113**	**0.488**	**0.141**
Oracle R-1	0.328	0.076	0.400	0.097	0.468	0.134	0.532	0.170
Oracle R-2	0.314	0.084	0.368	0.109	0.445	0.149	0.517	0.182

Length	50		100		200	
	R-1	R-2	R-1	R-2	R-1	R-2
ICSISumm	0.295	0.072	0.374	0.095	0.431	0.124
Greedy-KL	0.287	0.054	0.358	0.075	0.426	0.101
ProbSum	0.273	0.054	0.327	0.071	0.394	0.096
LLRSum	0.256	0.048	0.329	0.068	0.405	0.099
SumCombine	**0.309**	0.071	**0.382**	0.095	0.437	**0.117**
Oracle R-1	0.360	0.087	0.439	0.121	0.495	0.147
Oracle R-2	0.342	0.100	0.416	0.134	0.471	0.162

Table 6.10: Performance of systems on summaries of varied length on the DUC 2001 (top) and DUC 2002 (bottom) datasets. **Bold** and † represent statistical significant ($p < 0.05$) and close to significant ($0.05 \leq p < 0.1$) compared to ICSISumm (two-sided Wilcoxon signed-rank test).

	DUC 2005		DUC 2006		DUC 2007	
	R-1	R-2	R-1	R-2	R-1	R-2
ICSISumm	0.382	0.084	0.417	0.105	0.451	0.133
Greedy-KL	0.381	0.074	0.413	0.090	0.442	0.112
ProbSum	0.361	0.069	0.379	0.081	0.413	0.106
LLRSum	0.370	0.074	0.402	0.092	0.427	0.113
SumCombine	0.385	0.081	0.416	**0.099**	0.447	**0.123**
Oracle R-1	0.437	0.099	0.470	0.122	0.495	0.148
Oracle R-2	0.418	0.110	0.450	0.135	0.479	0.162

Table 6.11: Performance of systems on the DUC 2005–2007 datasets. **Bold** and † represent statistical significant ($p < 0.05$) and close to significant ($0.05 \leq p < 0.1$) compared to ICSISumm (two-sided Wilcoxon signed-rank test).

the result list. The top N summaries that can be expanded are kept and expanded next. The final candidate summaries are the top N in the result list, sorted by JS divergence in increasing order.

Experiments show that for both methods, the best possible performance (oracles) and final performance are lower than that of random selection. The main reason, I think, is the low diversity among the candidates. This is obviously true for the first method. For the beam search based method, most candidates are derived from a few branches of the search tree, a problem that also appeared in the k-best list generated by most systems in machine translation.[15] In contrast, the candidates generated by random selection are of high diversity.

Next, we provide insights on why our model failed as the summaries get longer. Our hypothesis is: *it becomes more difficult to outperform the best individual system as summaries get longer.* To test the hypothesis, we compute the percentage of candidate summaries that are better than (or at least as good as) ICSISumm for an input I, denoted as $r(I, L)$. Here L is the summary length. Then for each dataset, we compute the mean of $r(I, L)$ for all inputs, which measures the difficulty of outperforming ICSISumm while combining summaries of L words. When the basic summaries are longer than 100 words, the candidate summaries generated by random selection; we assume that the ROUGE score distribution among these candidate summaries is representative of all possible summaries that can be generated.

Figure 6.2 shows the percentage of candidate summaries that perform better than (or at least as good as) ICSISumm on the DUC 2001, 2002 data at different summary lengths. Both R-1 and R-2 are used to evaluate the performance. On the DUC 2001 dataset, system combination is obviously more difficult for longer summaries (i.e., 200, 400 words). On the DUC 2002 dataset, combining 50 word summary appears to be easier than combining 100 or 200 word summaries, while the trend is not so clear. In addition, we also observe that it is more difficult to beat ICSISumm in

[15] For methods that improve the diversity of sentences while decoding in MT, see Gimpel et al. (2013) and Liu and Huang (2014).

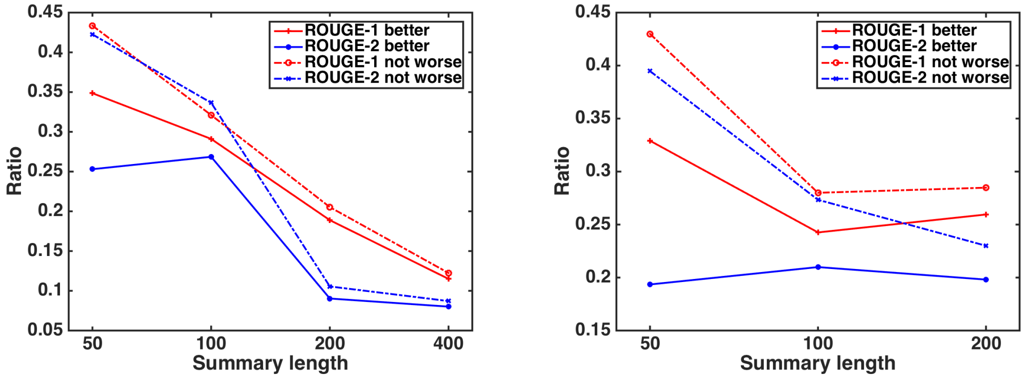


Figure 6.2: Percentage of summaries among the candidate summaries that outperform ICSISumm at different summary length (macro average). We show the trend on the DUC 2001 data (left) and the DUC 2002 data (right).

terms of R-2. This is consistent with our results that we can always achieve a better performance on R-1 than on R-2.

Overall, according to Figure 6.2, our hypothesis is correct most of the times. Here we give an explanation. The basic summarizers that we combined are of different performances. As can be seen in Table 6.10, the gap between good and mediocre systems becomes larger as the summaries get longer, an observation first discovered in Nenkova (2005). As a result, ICSISumm has higher ranks among the summary candidates as summaries get longer.

6.8 Conclusion

In this chapter, we present a new summary combination pipeline that includes two steps: (1) generating candidate summaries, and (2) selecting the best candidate. We use four portable unsupervised summarizers as the basic systems, which enables us to conduct a large scale study. Unlike prior work where the combined summaries are generated based on rankings of the input sentences that are assigned by different systems, we combine the summary outputs directly. Therefore, our method can be used to combine any summarization systems.

Based on the basic systems, we generate candidate summaries by combining whole sentences from the summaries produced by different systems. We show that the oracle choice among these candidates is much higher than the performance of the basic systems, which indicates high potential of summary combination.

We then propose a new supervised model to select among the candidate summaries. We propose a rich set of features that indicate the content quality of the entire summary. Specifically, the features are computed based on three sources: the input documents, the output of the basic systems, and the summary-article pairs from a large corpus. This is different from prior work in system combination where the indicator is only derived based on a single source. The effectiveness of different features are verified by ablation experiments on multiple datasets.

Our model is evaluated on nine DUC/TAC benchmarks. Experiments show that our model performs better than the basic systems while generating 50 and 100 word summaries, but does not outperform the best basic system while generating summaries of 200, 250 and 400 words. We empirically show that it is more difficult to generate high quality long summaries.

For future work, there are many avenues to explore. Our method generates candidate summaries by enumerating all possible combinations, which is relatively cumbersome. One may improve this by pruning some obviously wrong choices or performing beam search. Such efforts will also be helpful if one wants to use more systems or combine longer summaries. Moreover, none of the features used in our model are concerned with what words are used or what topics are discussed in a summary. It would be interesting to use vector representations of words (e.g., word2vec (Mikolov et al., 2013), neural network word embeddings (Turian et al., 2010)), and categorical features, all of which provide a dense representation of lexical information.

In the next chapter, we investigate how different factors of the input summaries or basic summarizers affect the final summary quality for system combination.

Chapter 7

Factors that Affect the Success of System Combination: An Empirical Study

In the previous chapter, we presented a method that combines the summary outputs (*basic summaries*) from a number of *basic summarizers (basic systems)*. In this chapter, we study how different factors affect the success of system combination.

This problem has been investigated by researchers in other domains of natural language processing. In machine translation (MT), one of the earliest study is conducted by Macherey and Och (2007). They show that it is important for the basic systems to have similar performance but high diversity. The importance of diversity has been confirmed by later work in MT (Devlin and Matsoukas, 2012; Cer et al., 2013; Gimpel et al., 2013). Apart from diversity, Gimpel et al. (2013) have demonstrated that it is easier to achieve an improvement when combining sentences with low translation qualities. They also show that having diverse input translations is helpful in this scenario. In addition to MT, prior work in automatic speech recognition has also showed that it is "*more effective to use models that have similar* *performance but are as diverse as possible, models that use different features, and different modeling approaches*" (Lee et al., 2014).

Inspired by previous research in other domains, we investigate whether similar results hold for summarization. Based on the data described in Chapter 6, we first conduct an exploratory study that focuses on properties of the basic summaries (micro level, Section 7.1). We then conduct a comprehensive study that focuses on properties of the basic systems (macro level, Section 7.2). **If you want to know the main results, read Section 7.2 first**. Section 7.3 includes a conclusion.

Here we define some terminology. We call the top-performing system *primary system*, the other systems *auxiliary systems*. System combination is *successful* if the combined system outperforms the primary system. We regard ROUGE-2 (R-2) as our *main metric* and also report the results on ROUGE-1 (R-1) when necessary. Section 2.2.2 describes why we make choices like this.

7.1 Summary Level Study

7.1.1 Introduction

We study how different properties of the *basic summaries* affect the success of system combination. Since we only use one set of the basic systems (e.g. the ones described in Chapter 6), we regard this study as an exploratory study. We focus on the following three factors: bigram overlap between the basic summaries (denoted as B),mean quality of the basic summaries (denoted as $\bar{Q}$), and difference in quality between the basic summaries (denoted as ΔQ). We test whether these factors affect three outcomes: quality of the oracle summary (denoted as O)[1]; potential improvement, measured by the relative improvement of oracle over the summary generated by the primary system (denoted as ΔO); real improvement, measured by the difference in

[1] Here oracle summary is the one with the highest ROUGE-2 among the candidate summaries. See Section 6.4 on how candidate summaries are generated.

summary quality between the combined system and the primary system (denoted as ΔY). In statistics, B, $\bar{Q}$, ΔQ are called *explanatory variables*; O, ΔO, ΔY are called *response variables*. We do not use the terms *independent variables* and *dependent variables*, because the input variables might be correlated.

Our study includes the following four experiments:

First, we study the relationship between explanatory variables (Section 7.1.2). Based on our basic systems, we find that: (1) lower diversity is correlated with higher mean quality of the basic summaries, (2) the correlation between diversity and the difference in quality between summaries is insignificant, (3) if the summaries have similar quality, then they tend to have low quality. We also discuss what can be learned from these observations. Since these factors are correlated, we compute both zero-order and partial correlations while studying the effects of these factors on response variables.

Second, based on the basic systems that we use, we investigate how the explanatory variables affect the quality of the oracle summary (Section 7.1.4). We observe that higher diversity (i.e., lower bigram overlap) is correlated with lower oracle if no factor is controlled, but correlated with higher oracle is the other factors are controlled (by partial correlation). Hence, suppose we have a set of summaries with similar performance, it is preferable to pick the summaries that are diverse.

Third, based on our systems, we investigate how these explanatory variables affect the potential of system combination (Section 7.1.5). By partial correlation, we find that larger potential is correlated with higher diversity, higher mean summary quality, and smaller difference in quality between summaries.

Finally, we study how these explanatory variables affect the real improvement (Section 7.1.6). We use the combination model proposed in Chapter 6. Here we can only observe a significant correlation between the similarity in summary quality and the real improvement. We will show in Section 7.2 that this observation also holds on the system level: it is superior to use systems that have similar performance.

Note that our study uses one combination method, based on one selection of the basic systems (the setting used in Chapter 6). Because of this, we cannot claim that our observations hold in general. Rather, we consider this section as a preliminary study. We will give a more comprehensive study in Section 7.2, where we use different sets of the basic systems and different combination methods.

7.1.2 Explanatory Variables

We use the basic summaries described in Section 6.3, which are generated by four portable unsupervised systems (ICSISumm, KL, FreqSum, TsSum) on the DUC 01–04 and TAC 08, 09 data (see Section 6.4). This includes 261 input sets. The primary system (best performing system) is ICSISumm.

We consider three properties of the basic summaries as our explanatory variables. These properties are shown to be correlated with the final translation quality for system combination in MT (see the second paragraph of this chapter for related work). First, we consider diversity, computed based on the mean bigram overlap between all pairs of basic summaries of an input (denoted as B). Let $S_1, \ldots, S_k$ denote the basic summaries of an input, B is computed as:

$$\frac{2}{k(k-1)} \sum_{1 \le i < j \le k} Jac(S_i, S_j) = \frac{2}{k(k-1)} \sum_{1 \le i < j \le k} \frac{|b(S_i) \bigcap b(S_j)|}{|b(S_i) \bigcup b(S_j)|} \tag{7.1}$$

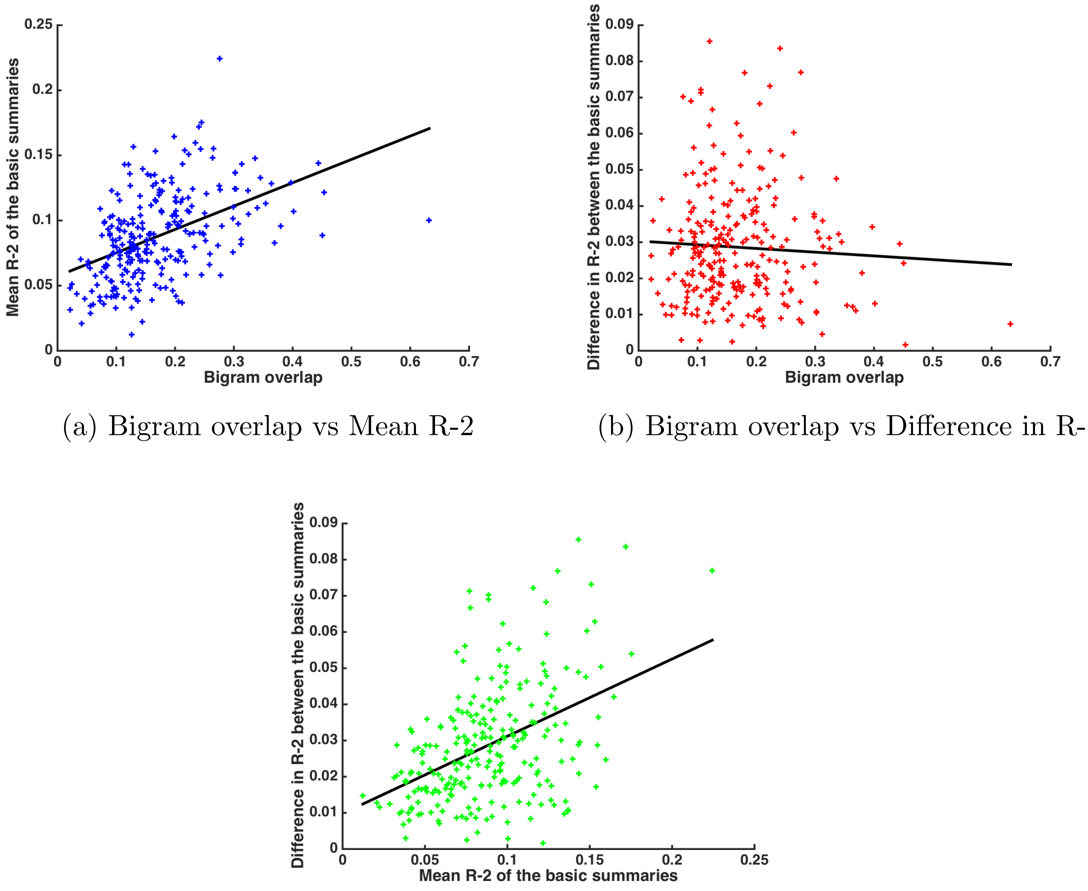
where $b(S_i)$ is the set of unique bigrams in summary S_i.[2] Second, we consider *mean quality* (ROUGE-2, R-2) of the basic summaries of an input (denoted as $\bar{Q}$). Third, we consider *difference in quality* between the basic summaries, which equals to the mean of differences in R-2 between all pairs of the basic summaries of an input (denoted as ΔQ). We use Spearman correlation rather than Pearson correlation, because Spearman correlation is more appropriate if we are unsure whether or not the relation is linear (Zou et al., 2003). Spearman correlation is used throughout this chapter.

[2] We perform stemming and include stopwords. Bigrams that include punctuations are excluded.

Table 7.1 shows the correlation between explanatory variables. The correlation is significant between B and $\bar{Q}$, $\bar{Q}$ and ΔQ, but not B and ΔQ. The trend can be seen clearly in Figure 7.1.

	B vs $\bar{Q}$	B vs ΔQ	$\bar{Q}$ vs ΔQ
Correlation	**0.509**	-0.014	**0.418**

Table 7.1: Spearman correlation between the explanatory variables. B: mean bigram overlap between the basic summaries, $\bar{Q}$: mean R-2 of the basic summaries, ΔQ: difference in R-2 between the basic summaries. **Bold** indicates statistical significant ($p < 0.05$).



(a) Bigram overlap vs Mean R-2

(b) Bigram overlap vs Difference in R-2

(c) Mean R-2 vs Difference in R-2

Figure 7.1: Scatter plots that show the correlation between the explanatory variables.

Our results have the following implications. First, based on B vs $\bar{Q}$, we know that if the machine summaries are similar, then these summaries tend to have high quality. If this observation can be generalized, then we have a method of estimating the difficulty of summarizing an input (this problem is studied in Nenkova and Louis (2008)). One can first use different systems to produce summaries, then compute the overlap between them. If the summaries have low overlap, this means the machine summaries are likely to have low quality, which suggests that the input is difficult to be summarized. Second, in our data, difference in quality between summaries (ΔQ) is independent of diversity (B). Hence, having equally good summaries does not imply these summaries choose either the same or diverse content. Third, in our data, average quality ($\bar{Q}$) is positively correlated with difference in quality (ΔQ), which suggests that it is more common for systems to generate equally bad rather than equally good summaries. Again, these findings are based on the four unsupervised systems introduced in Chapter 6.

7.1.3 Experimental Design

We use the two-step pipeline presented in Chapter 6 as our combination model (SumCombine). At the first step, we produce candidate summaries by combining whole sentences from the summaries generated by different systems. At the second step, we employ a supervised model to select among the candidates. The oracle summary is the one with the highest ROUGE-2 (R-2) among these candidates. ICSISumm is regarded as our primary system, because it has the highest performance among basic systems. Our experiment studies how different factors affect: (1) the quality of the summary (denoted as O), (2) the advantage of oracle over ICSISumm (denoted as ΔO), and (3) the advantage of SumCombine over ICSISumm (denoted as ΔY). We use R-2 to approximate the summary quality.

Since the input variables (factors) are correlated, it is necessary to control the other two factors while computing the correlation. Hence, we compute zero-order correlation (i.e., ignoring other factors) and partial correlation (consider the other factors) (Rummel, 1988). Our experiments answer the following two questions:

Zero-order correlation: Will the current explanatory variable (e.g., B) be correlated with the response variable (e.g., O), if the other variables (e.g., $\bar{Q}$ and ΔQ) are not controlled?

Paritial correlation: Will the current explanatory variable (e.g., B) be correlated with the response variable (e.g., O), if the other variables (e.g., $\bar{Q}$ and ΔQ) are controlled?

7.1.4 Effects on Oracle Performance

We study how the explanatory variables affect the oracle performance O (Table 7.2). We first discuss zero-order correlations. Not surprisingly, $\bar{Q}$ and O is highly correlated. A counter-intuitive result appears for the diversity factor, where lower diversity suggests a higher oracle. As B and $\bar{Q}$ is positively correlated, it is possible that B and O is positively correlated because $\bar{Q}$ and O is positively correlated.

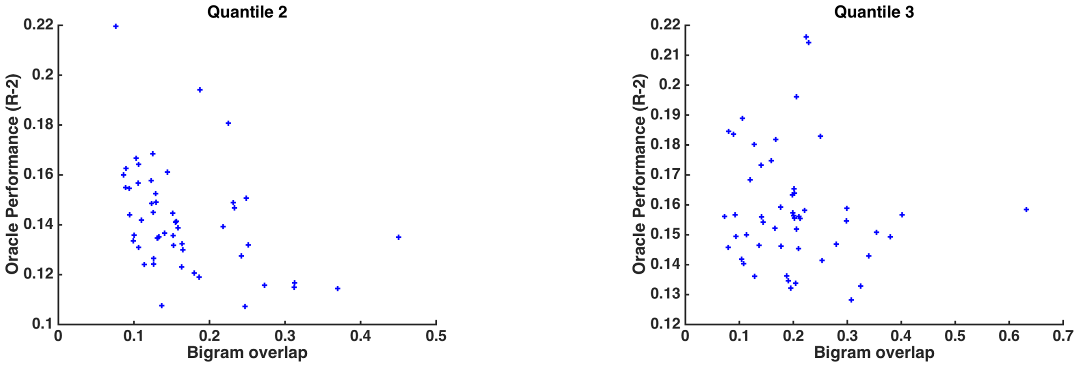
	Bigram overlap (B)	Mean R-2 ($\bar{Q}$)	Difference in R-2 (ΔQ)
Zero-order correlation	**0.327**	**0.889**	**0.565**
Partial correlation	**-0.216**	**0.857**	**0.447**

Table 7.2: Zero-order and partial Spearman correlation between the explanatory variables and the oracle performance. **Bold** indicates statistical significant ($p < 0.05$).

The result of partial correlation confirms our hypothesis—the correlation between B and O is negative. To better show this, we split the data points into 5-quantiles based on their mean performance, then compute the correlation between B and O for each quantile. Here we do not control for the difference in R-2 between systems (ΔQ), because the correlation between B and ΔQ is close to 0. As can be seen in Table 7.3 and Figure 7.2, all correlations are negative. Hence, if the quality of the basic summaries is controlled, then higher diversity implies higher oracle. Our experiment also stresses the importance of controlling factors while studying the relationship between explanatory and response variables.

	Quantile 0	Quantile 1	Quantile 2	Quantile 3	Quantile 4
R-2 range	[0.012, 0.058)	[0.058,0.076)	[0.076,0.095)	[0.095,0.117)	[0.117,0.224)
Correlation	-0.11	-0.17	**-0.48**	-0.11	**-0.37**

Table 7.3: Spearman correlation between bigram overlap and the oracle performance, after breaking down into 5-quantiles by mean R-2 of basic summaries. **Bold** indicates statistical significant ($p < 0.05$).



(a) Correlation = -0.48.

(b) Correlation = -0.11.

Figure 7.2: Spearman correlation between bigram overlap and oracle performance, after breaking down into quantiles by mean R-2 of basic summaries. We show the scatter plot for two quantiles.

7.1.5 Effects on Potential Improvement

Next, we look into how these factors affect the potential improvement, measured by the advantage of oracle over ICSISumm (ΔO). As shown in Table 7.4, no factor is significant for zero-order correlation, while all factors are significant for partial correlation.

Our result of partial correlation implies the following things. First, diversity is positively correlated with potential improvement. This is intuitive, because having diverse basic summaries is helpful to generate diverse candidate summaries. Moreover, as the basic summaries have low overlap with each other, there are more unique sentences used in all summaries, which will cause more summaries to be generated. Both reasons are helpful to improve the oracle performance. Second, mean quality is positively correlated with potential improvement. This is interesting, because we will later show in Section 7.2.1 that it is in fact difficult to achieve large real improvements while combining top-performing basic systems. Third, it is preferable for the summaries to have similar quality. Indeed, suppose one summary has higher quality than others (which is likely to be generated by the primary system), then it is difficult to achieve a large gain over that summary.

	Bigram overlap (B)	Mean R-2 ($\bar{Q}$)	Difference in R-2 (ΔQ)
Zero-order correlation	-0.055	0.085	-0.100
Partial correlation	**-0.162**	**0.204**	**-0.186**

Table 7.4: Zero-order and partial Spearman correlation between the explanatory variables and the advantage of oracle over ICSISumm. **Bold** indicates statistical significant ($p < 0.05$).

7.1.6 Effects on Real Improvement

Finally, we study how different factors affect the *success* of system combination, measured by the improvement of a combination system over the primary system on an input (denoted as ΔY). Here we use the summaries generated by SumCombine. We don't consider other approaches, because their performance falls behind that of SumCombine by a large margin (see Figure 6.1).

As can be seen in Table 7.5, the conclusion is very similar for zero-order and partial correlation. For diversity, the correlation is close to 0. Hence, though improving diversity leads to higher oracle and higher potential improvement, the potential is not exploited by SumCombine, based on our current input basic systems. For mean R-2, the correlation is also very weak. For difference in R-2, we observe a negative correlation, which implies that it is preferred for the basic summaries to have a similar quality. Moreover, we will show in Section 7.2.1 that this observation can be generalized into a more practical setting: it is also critical for the basic systems to have similar performance. Figure 7.3 plots the correlation between the explanatory variables and ΔY.

	Bigram overlap (B)	Mean R-2 ($\bar{Q}$)	Difference in R-2 (ΔQ)
Zero-order correlation	0.005	-0.017	**-0.153**
Partial correlation	-0.029	0.057	**-0.162**

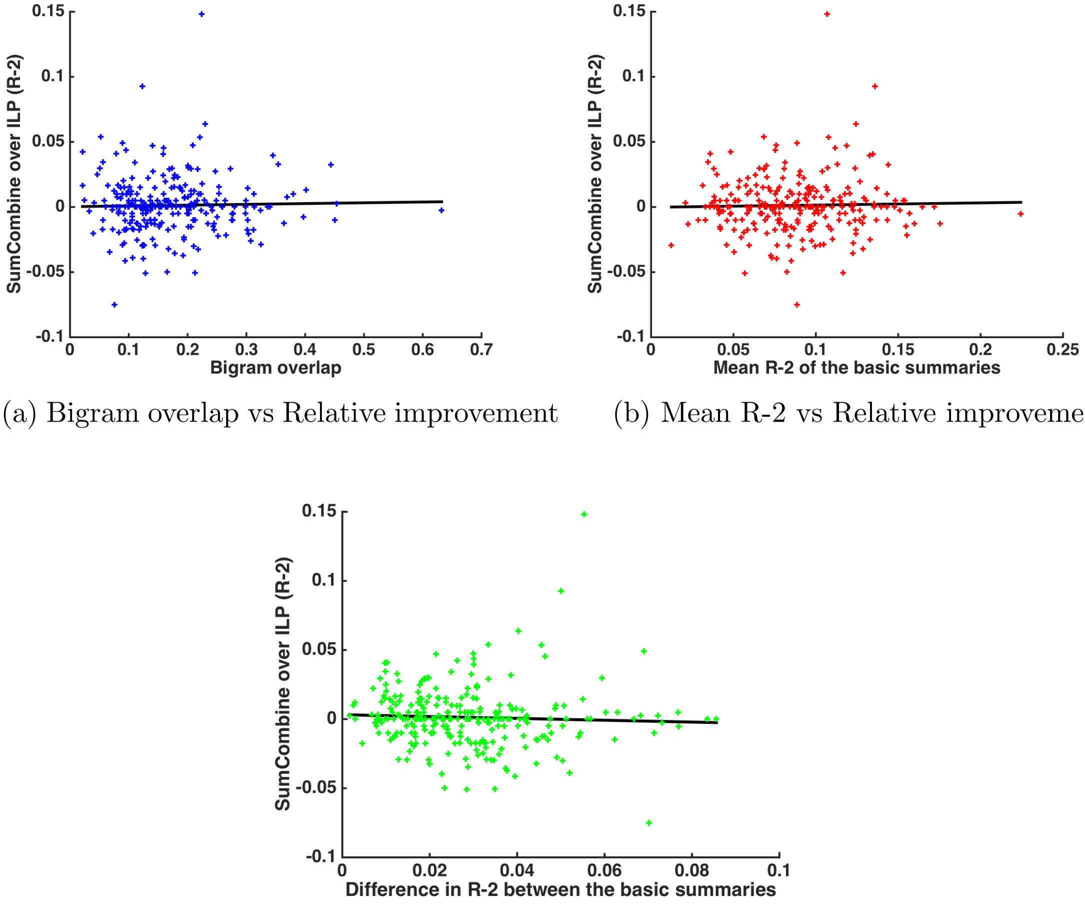
Table 7.5: Zero-order and partial correlation between the explanatory variables and the advantage of SumCombine over ICSISumm. **Bold** indicates statistical significant ($p < 0.05$).

7.2 System Level Study

7.2.1 Introduction

In Section 7.1, we have focused on properties of the basic summaries. In this section, we focus on properties of the systems that are combined (a.k.a, *basic systems*). Oftentimes, people need to predict whether or not system combination is necessary on the system level. They also need to decide which combination method should be used. Therefore, compared to the study in the last section, this study is of more practical meaning.

We mainly focus on *performance* of the basic systems. Prior work has shown that it is important for the basic systems to have similar performance in machine translation (MT) (Gimpel et al., 2013) and automatic speech recognition (ASR) (Lee



(a) Bigram overlap vs Relative improvement

(b) Mean R-2 vs Relative improvement

(c) Difference in R-2 vs Relative improvement

Figure 7.3: Correlation between the explanatory variables and the relative improvement.

et al., 2014). We wonder whether this is true for automatic summarization. Our preliminary study in Section 7.1 shows that it is preferred for summaries to have a similar quality.

For this study, we need a large number of different systems. Therefore, we conduct our experiments on the datasets released by the TAC 2008 shared task, which includes summaries generated by 72 systems. We experiment with 24 different combinations of the basic systems. In order to keep our conclusion unbiased towards a specific method, we use four different combination methods. Our experiments reveal the following findings:

First, it is important for the basic systems to have similar performance. To show this, we fix the primary system and change the selections of the auxiliary systems. Our experiment shows a negative correlation between *difference in ROUGE-2 between the primary system and mean of the auxiliary systems* and *ROUGE-2 of the combined systems*. Moreover, we show that the correlation is more significant for some combination methods than for others; we discuss why this happens.

Second, if the basic systems perform similarly, then even simple baseline methods can outperform the primary system. Moreover, our best model achieves a large gain (about 10% relative improvement) over the primary system in terms of ROUGE-2. This suggests that combination is very promising in this scenario.

Third, similar to MT (Gimpel et al., 2013), we find that combining low-performing systems is more likely to lead to larger improvement compared to combining top-performing systems. Hence, one should be careful when observing a large improvement while combining low-performing systems: the real improvement of combining top-performing systems might be much smaller.

Fourth, we compare four combination methods and discuss their effectiveness under different conditions. The supervised model (SumCombine) proposed in Chapter 6 performs the best. Input-summary Jensen-Shannon divergence performs the best among the three baselines, which is sometimes on par with (or even better than) SumCombine. The other two baselines take advantage of consensus between summaries, which only gives a good performance when the basic systems perform similarly. These results confirm our conclusion in Chapter 6 that it is necessary to use signals from the input and basic summaries, with the input being more important.

Our experiments are mainly concerned with *performance* of the basic systems. It is true that many factors could affect the success of system combination, such as properties of the input, combination method, and diversity between the basic systems. Our experiments are conducted using the same class of methods, based on the same input, which controls the first two factors.

Note that we do not control the diversity between systems. Even though the summaries from different systems can be of different diversity on the summary level, the degree of bigram overlap on the system level (which takes the mean of bigram overlap on the summary level) falls into a not-so-large range: between 0.077 and 0.222 measured by Jaccard coefficient (see Table 7.9, row Bi-Jac).[3] Hence, we make an assumption that diversity will not affect the final performance much on the system level. Moreover, we have shown that diversity does not affect the relative improvement of system combination on the summary level in Section 7.1, which suggests that it is safe to make this assumption.

Below we first describe our dataset (Section 7.2.2). We then describe the combination methods (Section 7.2.3) and results (Section 7.2.4).

7.2.2 Data

We use the summaries collected from the TAC 2008 shared task (Dang and Owczarzak, 2008). There were 33 teams participated in TAC 2008. Among them, 14 teams submitted three systems and 10 teams submitted two systems. This resulted in a total of 72 systems.[4] Figure 7.4 shows the histogram in terms of ROUGE-2 of these systems.

Summaries generated by systems submitted by the same team might be very similar or even identical. Clearly, we do not want to combine these systems. Hence, when this happens, we only include the system with a higher ROUGE-2 (R-2). Here we regard two systems as *near identical* if the Jaccard coefficient on the word level between these systems is greater than 0.6 (see Section 4.2.4 on how this is computed).[5] There are 50 systems left after removing the systems that are near identical. The new ID of these systems are assigned based on their ranks in terms of R-2 (see Table 7.6).

[3] We perform stemming and include stopwords. Bigram that includes punctuations are excluded.

[4] A baseline system that extracts the first 100 words from the latest document is also included.

[5] with stemming, including stopwords, excluding punctuations

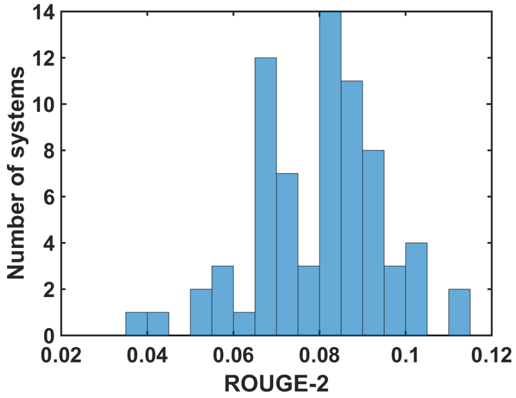


Figure 7.4: ROUGE-2 of systems on the TAC 2008 shared task. y-axis is the number of systems within one performance range.

ID	R-2	ID	R-2	ID	R-2	ID	R-2	ID	R-2
1	0.1113	11	0.0905	21	0.0823	31	0.0715	41	0.0670
2	0.1104	12	0.0885	22	0.0811	32	0.0708	42	0.0645
3	0.1036	13	0.0881	23	0.0800	33	0.0698	43	0.0624
4	0.1000	14	0.0880	24	0.0795	34	0.0698	44	0.0582
5	0.0992	15	0.0857	25	0.0783	35	0.0690	45	0.0576
6	0.0951	16	0.0856	26	0.0782	36	0.0684	46	0.0559
7	0.0948	17	0.0846	27	0.0768	37	0.0676	47	0.0527
8	0.0941	18	0.0840	28	0.0745	38	0.0676	48	0.0506
9	0.0934	19	0.0839	29	0.0734	39	0.0676	49	0.0424
10	0.0917	20	0.0825	30	0.0731	40	0.0675	50	0.0392

Table 7.6: ROUGE-2 (R-2) of the systems on the TAC 2008 shared task. If two systems are too similar, then only one of them is preserved.

7.2.3 Combination Methods

We use four combination methods (Greedy-KL, Pseudo, JS, SumCombine) for our experiments. The first method directly uses the Greedy-KL summarizer over the basic summaries. The other three are based on the two-step pipeline presented in Chapter 6: they employ the same method to generate candidate summaries (see Section 6.4) but different methods to select among these candidates (denoted as $C_1 \dots C_n$).

Because the first three methods are unsupervised and easy-to-implement, we regard them as baselines. In Chapter 6, we have shown that SumCombine outperforms Greedy-KL and JS significantly, where we combine four unsupervised basic systems.

Greedy-KL: We regard system combination as another summarization task. We run the Greedy-KL summarizer (Section 3.5.3), which iteratively picks a sentence that minimizes the Kullback-Leibler (KL) divergence between the current summary and the basic summaries. Our experiments in Section 6.7.2 have shown that using Greedy-KL is better than using ICSISumm (Figure 6.1).

Pseudo: Pseudo selects the C_i with the highest ROUGE score, using the basic summaries as pseudo-references. Similar approach has been shown effective in system combination for machine translation (MT), where the translation with the highest BLEU score compared to pseudo-references is selected (Macherey and Och, 2007). This method is also used in automatic evaluation without human references for machine translation and automatic summarization.

JS: JS selects the C_i with the smallest Jensen-Shannon (JS) divergence between C_i and the documents to be summarized (i.e., *input*). JS is the best baseline method among the ones we proposed in Section 6.6. Input-summary JS-divergence is also the best single metric in identifying a better summarizer without human models (Louis and Nenkova, 2009).

SumCombine: This is our main method in Chapter 6. It selects the best C_i based on a rich set of features. The same as Chapter 6, our model is still trained on the DUC 03 and 04 dataset, where the candidate summaries (training instances) are generated by combining four portable unsupervised systems (see Section 6.4). ROUGE-1 is used for labeling the training instances. Note that because we are not combining summaries from the same set of systems while training and testing, system identity features cannot be used (this class of features suggests which summarizer is better). All other features are used in our model.

Using multiple methods gives us three advantages: (1) our conclusions will not be made based on a specific method, (2) it is helpful to know if certain methods work better under certain conditions, and (3) it helps us to understand what properties do we prefer combination methods to have. Table 7.7 compares these methods.

	Greedy-KL	Pseudo	JS	SumCombine
Supervised?	No	No	No	Yes
Need candidate summaries?	No	Yes	Yes	Yes
Signal from the input?	No	No	Yes	Yes
Signal from the basic summaries?	Yes	Yes	No	Yes
Signal from the NYT corpus?	No	No	No	Yes

Table 7.7: Comparison between different combination methods in terms of strategies.

7.2.4 Experiments and Results

We experiment with 24 combinations of the basic systems and combine three systems at a time. Based on the questions that we want to answer, we classify the combinations into three groups, which corresponds to three groups of experiments (see Table 7.9). Below we describe our experiments and findings.

First, we look into the question: *"How important it is for the auxiliary systems to have a performance similar to the primary system?"* To answer this question, we fix the primary system and experiment with different selections of the auxiliary systems. Let Δ denote the difference in ROUGE-2 (R-2) between the primary system and mean of the auxiliary systems, we have the hypothesis that as Δ gets larger, our combined system performs worse. Here we conduct two experiments. For the first experiment (*Exp 1*), we use System 1 as the primary system, because it has the best performance. For the second experiment (*Exp 2*), we use System 6 as the primary system, because Systems 6, 7, and 8 have very similar performances, which yields a very small Δ (0.0007). We choose eight combinations of the basic systems for both

Exp1 and Exp 2. Table 7.9 shows the final performance for all four combination approaches.

Table 7.8 shows the Spearman correlation between Δ and the performance of combination. Not surprisingly, the correlation is always negative. Even though we only have eight observations, the correlation is significant or close to significant for KL and Pseudo. For JS, the correlation is insignificant. For SumCombine, the correlation is insignificant for Exp 1 but significant for Exp 2. Indeed, as KL and Pseudo take advantage of consensus between summaries, the existence of poor systems is likely to cause a devastating affect. JS estimates the importance of summaries solely based on the input. Therefore, what we choose as basic summaries only affect what candidate summaries are generated, but not the summary scoring process. SumCombine uses indicators from the input and consensus between the basic summaries.

	Greedy-KL	Pseudo	JS	SumCombine
Exp 1	**-0.976**	**-0.928**	-0.238	-0.131
Exp 2	-0.690†	-0.667†	-0.262	**-0.738**

Table 7.8: Spearman correlation between Δ and the performance of different combination methods for Exp 1 and Exp 2. **Bold** and † indicate statistical significant ($p < 0.05$) and close to significant ($0.05 \leq p < 0.1$).

It is worth noting that SumCombine and JS still outperform the primary system most of the times, even if the auxiliary systems have a low R-2. For example, among all 16 combinations in Exp 1 and Exp 2, SumCombine achieves a higher ROUGE-2 than the primary system 15 times, except for the case when $\Delta = 0.0356$. If we look into the ROUGE-1, JS and SumCombine can always outperform the primary system (see Table 7.11, which is at the end of this chapter). Hence, for JS and SumCombine, our experiment in Table 7.8 actually shows that as the auxiliary systems get better, the combined system tends to have larger advantages, especially for SumCombine. The other two baselines (Greedy-KL, Pseudo) cannot outperform the primary system when the auxiliary systems have a low performance ($\Delta \geq 0.02$).

Exp 1	Basic systems				Combination methods					
	Sys1	Sys2	Sys3	Δ	KL	Pseudo	JS	SumCombine	Oracle	Bi-Jac
1, 2, 3	.111	.110	.103	.0045	.110	**.119**	**.111**	**.118**	**.147**	0.222
1, 3, 4	.111	.103	.100	.0097	.108	**.113**	**.115**	**.113**	**.142**	0.157
1, 4, 5	.111	.100	.099	.0119	.109	.111	**.112**	**.113**	**.139**	0.189
1, 4, 10	.111	.100	.092	.0155	.102	.102	.109	**.114**	**.140**	0.141
1, 2, 25	.111	.110	.078	.0171	.100	**.112**	**.115**	**.114**	**.132**	0.201
1, 10, 11	.111	.092	.090	.0202	.100	.107	**.118**	**.113**	**.146**	0.160
1, 15, 16	.111	.086	.086	.0256	.098	.103	.109	**.120**	**.142**	0.147
1, 27, 28	.111	.077	.075	.0356	.093	.100	.109	.108	**.146**	0.085
Mean	.111	.097	.090	.0170	.102	.109	**.113**	**.114**	**.142**	0.163

Exp 2	Basic systems				Combination methods					
	Sys1	Sys2	Sys3	Δ	KL	Pseudo	JS	SumCombine	Oracle	Bi-Jac
6, 7, 8	.095	.095	.094	.0006	**.097**	**.100**	**.104**	**.108**	**.134**	0.178
6, 10, 11	.095	.092	.090	.0039	**.103**	**.101**	**.103**	**.107**	**.131**	0.187
6, 10, 15	.095	.092	.086	.0061	**.097**	**.095**	**.104**	**.110**	**.133**	0.150
6, 15, 16	.095	.086	.086	.0093	**.100**	**.097**	**.102**	**.108**	**.128**	0.186
6, 7, 28	.095	.095	.075	.0104	**.097**	.094	**.105**	**.102**	**.136**	0.110
6, 15, 28	.095	.086	.075	.0148	**.096**	**.097**	**.106**	**.107**	**.135**	0.101
6, 27, 28	.095	.077	.075	.0193	.091	**.096**	**.103**	**.103**	**.137**	0.077
6, 33, 34	.095	.070	.070	.0252	.083	.086	**.096**	**.102**	**.125**	0.080
Mean	.095	.086	.081	.0110	.095	.096	.103	.106	**.133**	0.134

Exp 3	Basic systems				Combination methods					
	Sys1	Sys2	Sys3	Δ	KL	Pseudo	JS	SumCombine	Oracle	Bi-Jac
1, 2, 3	.111	.110	.103	.0045	.110	**.119**	**.111**	**.118**	**.147**	0.222
2, 3, 4	.110	.103	.100	.0087	.109	**.111**	.110	.110	**.136**	0.176
3, 4, 5	.103	.100	.099	.0040	**.105**	**.106**	**.111**	**.111**	**.135**	0.143
4, 5, 6	.100	.099	.095	.0028	**.101**	**.105**	**.106**	**.111**	**.132**	0.151
5, 6, 7	.099	.095	.095	.0042	**.100**	**.103**	**.109**	**.114**	**.142**	0.133
6, 7, 8	.095	.095	.094	.0006	**.097**	**.100**	**.104**	**.108**	**.134**	0.178
7, 8, 9	.095	.094	.094	.0009	**.102**	**.100**	**.098**	**.102**	**.129**	0.210
17, 18, 19	.085	.084	.084	.0006	**.092**	**.086**	**.098**	**.099**	**.123**	0.148
25, 26, 27	.078	.078	.077	.0008	**.085**	**.085**	**.090**	**.087**	**.110**	0.121
33, 34, 35	.070	.070	.069	.0004	**.078**	**.084**	**.087**	**.089**	**.112**	0.079
Mean	.095	.093	.091	.0027	**.098**	**.100**	**.102**	**.105**	**.130**	0.157

Table 7.9: ROUGE-2 of the basic systems and different combination methods. **Bold** indicates better than the primary system (Sys1). Bi-Jac is the bigram overlap between summaries, measured by Jaccard coefficient.

Second, we look into the question: *"If the basic systems have a similar performance, will all combination methods outperform the primary system comfortably?"* Our third experiment (Exp 3) addresses this question, where we combine systems that are ranked next to each other in the TAC 08 workshops.

Among our four methods, Greedy-KL is the simplest, because no candidate summary needs to be generated. This method has the smallest gain over the primary system (+0.003). We achieve larger gains using the other two baselines (GreedyKL, Pseudo), where the improvements are 0.005, 0.007 respectively. The most sophisticated method (SumCombine) achieves the largest gain over the primary system (+0.01). Hence, when the basic systems have similar performance, system combination is very promising: people may choose very simple methods, or they may use more sophisticated methods to produce better summaries.

Generally speaking, methods that estimate the importance based on consensus of basic summaries (Greedy-KL, Pseudo) are effective only when the summaries are of similar qualities. These methods are unlikely to perform well when the systems have large gaps in performance (for example, some data points in Exp1 and Exp2). This claim is also supported by our results in the last chapter (Figure 6.1). Approaches based on input-summary JS divergence has a strong performance even when the auxiliary systems do not perform as well as the primary system. Thus, we recommend this model if one wants a tradeoff between complexity and performance. SumCombine performs the best, which confirms the effectiveness of using various features derived from different resources while combination. In fact, if the system identity features can be used, the performance is likely to be improved, as evidenced by our feature ablation experiments in Table 6.9.

Note that the relative rankings of our four methods are different while combining different systems. Therefore, when evaluating combination methods, it is better to experiment with different selections of the basic systems (this section is an example).

Exp 3 suggests another claim: *the advantage of system combination can be better* *exploited if we combine low-performing systems.* For example, while combining system 33, 34, and 35, SumCombine outperforms the primary system by 0.019; while combining system 1, 2, and 3, SumCombine only outperforms the primary system by 0.007. In order to more concretely show this, for the data points in Exp 3, we compute the correlation between mean R-2 of the basic systems and relative improvement of our methods over the primary system (Sys1). As can be seen in Table 7.10, the correlation is positive for four methods and significant for three methods. Another observation that supports this claim is: the improvement in Exp 2 is larger than Exp 1. This claim suggests that if we only have low-performing or mediocre systems, then system combination is likely to result in a large improvement. However, it is unclear whether combining top-performing systems will reach the ceiling at some point—the advantage of SumCombine over system 1 while combining system 1, 2, and 3 is moderate and the potential (as evidenced by the column Oracle in Table 7.9, 7.11) is still very high.

One important thing can be learned from the claim above. If one method achieves 0.02 absolute improvement over the primary system and the primary system does not perform strongly, we need to be skeptical on how good this method is. Improving over low-performing systems is very easy. It is more convincing if a method could outperform the primary system by a considerable margin and also advance the state-of-the-art by combining top-performing systems.

	Greedy-KL	Pseudo	JS	SumCombine
Correlation	**0.948**	0.345	**0.830**	**0.697**

Table 7.10: Spearman correlation between mean R-2 of the basic systems and relative improvement (R-2) of different combination methods for the basic systems in Exp 3. **Bold** indicates statistical significant ($p < 0.05$).

7.3 Conclusion

In this chapter, we answer the question: "*What factors could affect the success of system combination for multi-document summarization?*"

First, based on the systems and data in Chapter 6, we consider three properties that are concerned with the summaries we combined (i.e., *basic summaries*): diversity, mean quality and difference in quality between summaries. Experiments based on our input basic systems show that: (1) lower diversity is related to higher mean quality, (2) diversity is independent of difference in quality, and (3) similar quality between summaries is related to low quality of the summaries. We show that higher oracle is related to higher diversity, higher mean quality and larger difference in quality between systems. We also show that larger potential improvement is correlated with higher diversity, higher mean quality and similar quality between summaries. However, only the last factor is significantly correlated with the real improvement. Different from prior work in MT, we do not find a correlation between diversity and real improvement on our data. Note that our experiment is conducted using a single set of basic systems, based on one combination method. Future work may investigate whether these observations hold in general.

Second, we consider properties that are concerned with the systems that we combined (i.e., *basic systems*). This study is of more practical meaning, because ultimately we need to combine existing systems for new summarization problems (datasets). We experiment with four combination methods and 24 different selections of the basic systems. We show that: (1) System combination works better when the auxiliary systems perform better. (2) If the basic systems have similar performance, even simple combination methods can be successful. (3) Summarizing by using the summary with the smallest input-summary JS divergence (JS) only falls behind a variation of our best model by a small margin. Hence, we recommend JS as a strong baseline. (4) Our best model and JS also generate summaries better than the primary system while combining good and mediocre systems. (5) Signals from the input set is the most important. It is also critical to utilize consensus between summaries and global knowledge. (6) It is very easy to achieve a large improvement by combining low-performing systems.

Future work may focus on combining summaries from the state-of-the-art systems, where we believe summaries with better content quality can be generated.[6] Moreover, we find that diversity between basic summaries is correlated with qualities of the basic summaries (Section 7.1.2). If this finding can be generalized, then we have a method of predicting the difficulty of summarizing an input. One can first generate summaries from different systems, then compute the overlap between these summaries. If the machine summaries are diverse, then this implies that these machine summaries tend to have low quality, which implies that the input is more difficult to be summarized.

[6] We did not use the summaries from our repository in Section 4.2, which includes summaries from 12 state-of-the-art and baseline systems. This is because our SumCombine model is developed on the DUC 2003 and 2004 data. Using this dataset might risk overfitting. Moreover, the TAC 08 workshop has more participating systems than we have in our repository.

Exp 1	Basic systems			Combined systems					
	Sys1	Sys2	Sys3	KL	Pseudo	JS	SumCombine	Oracle	Bi-Jac
1, 2, 3	.383	.378	.390	**.389**	**.397**	**.394**	**.397**	**.445**	0.222
1, 3, 4	.383	.390	.371	**.387**	**.395**	**.394**	**.395**	**.436**	0.157
1, 4, 5	.383	.371	.365	.378	.383	**.389**	**.388**	**.426**	0.189
1, 4, 10	.383	.371	.355	.378	**.389**	**.394**	**.394**	**.426**	0.141
1, 2, 25	.383	.378	.331	.371	**.386**	**.389**	**.385**	**.419**	0.201
1, 10, 11	.383	.355	.346	.372	**.383**	**.397**	**.394**	**.436**	0.160
1, 15, 16	.383	.349	.342	.371	**.390**	**.389**	**.398**	**.438**	0.147
1, 27, 28	.383	.345	.334	.370	**.388**	**.397**	**.389**	**.442**	0.085
Mean	.383	.367	.364	.377	**.389**	**.393**	**.392**	**.434**	0.163

Exp 2	Basic systems			Combined systems					
	Sys1	Sys2	Sys3	KL	Pseudo	JS	SumCombine	Oracle	Bi-Jac
6, 7, 8	.376	.358	.371	.369	**.377**	**.388**	**.391**	**.427**	0.178
6, 10, 11	.376	.355	.346	**.379**	.376	**.383**	**.388**	**.423**	0.187
6, 10, 15	.376	.355	.349	.375	.372	**.389**	**.396**	**.431**	0.150
6, 15, 16	.376	.349	.342	**.377**	.375	**.388**	**.396**	**.428**	0.186
6, 7, 28	.376	.358	.334	.369	.375	**.385**	**.384**	**.427**	0.110
6, 15, 28	.376	.349	.334	.374	**.379**	**.391**	**.390**	**.434**	0.101
6, 27, 28	.376	.345	.334	.373	**.383**	**.391**	**.388**	**.434**	0.077
6, 33, 34	.376	.320	.336	.353	.369	**.382**	**.387**	**.425**	0.080
Mean	.376	.349	.343	.371	.376	**.387**	**.390**	**.429**	0.134

Exp 3	Basic systems			Combined systems					
	Sys1	Sys2	Sys3	KL	Pseudo	JS	SumCombine	Oracle	Bi-Jac
1, 2, 3	.383	.378	.390	**.389**	**.397**	**.394**	**.397**	**.445**	0.222
2, 3, 4	.378	.390	.371	**.388**	**.392**	**.390**	**.390**	**.428**	0.176
3, 4, 5	.390	.371	.365	.384	**.390**	**.395**	**.392**	**.432**	0.143
4, 5, 6	.371	.365	.376	**.377**	**.382**	**.385**	**.390**	**.428**	0.151
5, 6, 7	.365	.376	.358	**.372**	**.384**	**.390**	**.395**	**.441**	0.133
6, 7, 8	.376	.358	.371	369	**.377**	**.388**	**.391**	**.427**	0.178
7, 8, 9	.358	.371	.369	**.372**	**.382**	**.378**	**.384**	**.421**	0.210
17, 18, 19	.358	.355	.343	**.367**	**.366**	**.378**	**.383**	**.419**	0.148
25, 26, 37	.331	.321	.345	**.352**	**.348**	**.366**	**.360**	**.396**	0.127
33, 34, 35	.336	.326	.326	**.343**	**.357**	**.367**	**.364**	**.405**	0.079
Mean	.365	.361	.362	**.377**	**.383**	**.385**	**.387**	**.424**	0.157

Table 7.11: ROUGE-1 of the basic systems and different combination methods. **Bold** means better than the primary system (Sys1). Bi-Jac is the bigram overlap between summaries, measured by Jaccard coefficient.

Chapter 8

Conclusion and Future Work

In this thesis, we provide insights on how to improve content selection in multi-document summarization. In this concluding chapter, we reiterate our main contributions and discuss future directions.

8.1 Main Findings and Contributions

In **Chapter 3**, we study the problem of "*estimating the importance of words in the input documents*", a problem less studied compared to sentence importance estimation. We find that three popular unsupervised word weighting methods have low correlations in terms of predicted weights. Inspired by this, we explore and combine different indicators (features) of word importance. Apart from features that are traditionally used (frequency and location), we include features such as word properties (part-of-speech, named entity, capitalization), subjectivity, reinforcement from machine summaries, semantic categories and global importance of words. We find that there are more nouns and past tense verbs, fewer comparative adjectives and adverbs in human summaries. We also find that there are more words with the named entity tag of "*Organization*", "*Locations*" and more words that are capitalized in human summaries. Experiments show that apart from frequency and location, word property and global importance are the most important features.

To quantify the estimation of word importance, we introduce the task of identifying words used in human summaries (i.e., *summary keywords*). We also introduce the *Keyword Pyramid Method*, a new method that evaluates the identification of summary keywords. Experiments show that a logistic regression model that combines the features above outperforms several prior methods on this task.

In the first half of **Chapter 4**, we study how different word weighting methods affect the final performance in summarization. Based on a greedy extractive summarizer (GreedySum), we show that our best method of identifying summary keywords (our proposed model in Chapter 3) is also the best method for summarization, which generates summaries on par with the state-of-the-art systems on the DUC 2004 dataset. Hence, our proposed new model is effective in summarization. Based on GreedySum, we compare two strategies of assigning word importance: assigning binary or real-valued weights. The former (latter) strategy works better if the weights are estimated by unsupervised (supervised) methods. We hypothesize that this is because supervised models assign more appropriately scaled weights to words compared to unsupervised methods (see Table 4.2).

In the second half of **Chapter 4**, we construct a repository that includes summaries from six state-of-the-art and six baseline systems on the DUC 2004 dataset, the most widely used dataset for generic multi-document summarization. Our repository addresses the problem that prior systems were evaluated on different datasets, using different ROUGE metrics. Our repository also makes it feasible to use paired tests to compare significant differences in performance between systems, as recommended in Rankel et al. (2011). Experiments show that the state-of-the-art systems have similar performance but generate very diverse summaries. Inspired by this, we study system combination in Chapter 6.

Since its release, researchers have used our repository to compare with the systems they have developed (Banerjee et al., 2015; Mogren et al., 2015). Our repository has also facilitated research in summarization evaluation. Graham (2015) conducts experiments on our repository and advocates a metric variant of ROUGE, which is different from the metric we used. Note that the gold standard human score used in Graham (2015) combines coverage (content quality) and linguistic quality. In my opinion, if the goal is to identify a better automatic metric of evaluating content quality, it is better that *only* the manual content quality is used as the gold standard.

Most summarization systems estimate content importance only based on the input. In **Chapter 5**, we investigate the problem of "*using global knowledge for estimating content importance*", where we mine and apply knowledge independent of a particular input. We propose two methods of mining global knowledge. First, based on the hypothesis that words in certain categories are likely to be globally important, we mine such information from a subjectivity dictionary (MPQA) and a semantic dictionary (LIWC). We show that humans tend to avoid words of strong subjectivity and tend to include words that belong to the categories of *death*, *anger*, *achieve* and *negative emotions*. Second, based on 160K summary-article pairs from the NYT corpus, we directly estimate the global importance of words (i.e., *global indicators*). We show that a system that produces summaries only based on these indicators (i.e. do not conduct any analysis of the input) perform similar to a standard baseline, which selects the first k words from the lastest article.

To explain why certain words are identified as globally important (unimportant), we study the categories that these words belong to in MPQA and LIWC (Section 5.4.2). The following categories include more intrinsically important words than unimportant words: *money*, *death*, *work*, and *religion*.

We examine the effectiveness of global knowledge on four tasks. Including global knowledge leads to a small increase on summary keyword identification, summarization, and system combination. Remarkably, including global knowledge leads to a significant improvement in identifying words that have low frequency in the input.

The idea of using global knowledge has rejuvenated in the past two years. Our paper (Hong and Nenkova, 2014a) is among one of the papers that explore this idea (Wan and Zhang, 2014; Li et al., 2015; Nye and Nenkova, 2015; Cao et al., 2015b).

Chapter 6 describes a new pipeline of system combination for multi-document summarization (SumCombine). This pipeline includes two stages: generating candidate summaries and summary selection. In the first stage, we generate candidate summaries by combining whole sentences from the outputs produced by different systems (i.e., *basic systems*). We show that an oracle system that picks the best candidate performs much better than the basic systems, which implies that the potential of system combination is high. In the second stage, we present a support vector regression model that relies on a rich set of new features. Ablation experiment confirms the effectiveness of our proposed features. Our model (SumCombine) performs better than the basic systems and several baselines while combining short summaries, which is comparable to the state-of-the-art on multiple benchmarks. However, our model fails to generate better summaries when the summaries are long. We empirically show that it becomes more difficult to outperform the best basic system (i.e., *primary system*) as the summaries get longer (Section 6.7.4).

In **Chapter 7**, we study how different factors affect the final performance in system combination. Our main study focuses on properties of the basic systems (Section 7.2). We use four combination methods: SumCombine, a system that selects the summary with the smallest input-summary Jensen-Shannon divergence (JS), and two systems that use consensus between the systems. Our experiments show the following results. First, combination methods tend to have a greater advantage over the primary system when the systems combined have similar performance (Table 7.8). Second, when the systems have similar performance, all combination methods outperform the primary system; when the primary system performs much better than others, only SumCombine and JS outperform the primary system. Third, compared to combining high-performing systems, combining low-performing systems is easier to result in a larger improvement. Hence, if a combination method achieves a large improvement by combining straw-man systems, one needs to be skeptical. It is more convincing if a method outperforms the primary system by a considerable margin and also advance the state-of-the-art by combining top-performing systems.

We also conduct a study that focuses on properties of the summaries produced by different systems (Section 7.1). We regard this as a preliminary study, because we only use one selection of the basic systems and one combination method (*our settings in Chapter 6*). We are surprised to find that diversity does not help in improving the real performance, though it is helpful to improve the best possible performance. Future research may investigate whether this holds in general. Similar to Section 7.2, we observe that it is important to combine summaries of similar quality.

8.2 Future Work

Based on this thesis, we discuss four avenues for future research:

Improving the Estimation of Concept Importance

According to Gillick and Favre (2009), concepts can be defined as "*words, named entities, syntactic subtrees, semantic relations, etc.*" Among them, words and bigrams are the most widely used in summarization systems. Because of this, researchers have explored features that encode the importance of unigrams (Hong and Nenkova, 2014a) and bigrams (Li et al., 2013b; Li et al., 2015). Future work can investigate features that are helpful to identify other concepts used in human summaries. For example, we estimate the intrinsic importance of words based on the change in probability between the summaries and the input. This method can be extended to estimate the global importance of other concepts, such as named entities, production rules (e.g., NP → VB VP), or triples from dependency parse (e.g., *det*(book, the)).

Improving the estimation of concept importance might benefit summarization systems that rely on deeper levels of linguistic analysis, such as: discourse, syntactic, and semantic. One recent paper shows that in the domain of Wikipedia and legal text summarization, maximizing the coverage of semantic and syntactic concepts is better than maximizing the coverage of bigrams (Schluter and Søgaard, 2015). Estimating concept importance in a discriminative way might benefit their system, where the weights were estimated in an unsupervised fashion.

Improving the System Combination Model

Our summary combination pipeline includes two steps: generating candidate summaries and selecting the summary with the highest quality. There is room for improvement for both steps. During the first step, we exhaustively enumerate all possible combinations. This might be improved by pruning some obviously low quality summaries or performing beam search. Such efforts will also be helpful if one wants to use more systems or combine longer summaries. For the second step, none of the features we used are concerned with which words or topics are used in a summary. Future work may investigate whether or not incorporating such information will be helpful. Possible options include lexical features, dictionary features, and word embeddings learned from neural networks.

Even though our model outperforms the basic systems, the gap in performance between the oracle system and our combination model is still large. Hence, there is high potential that combination methods can be improved. Moreover, it is very promising to combine summaries from the state-of-the-art systems: the final performance is likely to be better than the one we report in Chapter 6.

A Decoding and Reranking Model for Summarization

Our two-stage system combination pipeline can be extended for summarization. In the first stage (i.e., decoding), candidate summaries can be generated based on the input. In the second stage (i.e., reranking), we train a model to identify the best summary among these candidates. Using a discriminative model to select among the K-best candidates has been shown very useful in parsing (Collins and Koo, 2005; Charniak and Johnson, 2005) and machine translation (Shen et al., 2004).

Our proposed method is different from previous summarization systems in the following ways. First, different from methods that iteratively select an important sentence, we directly optimize the quality of the entire summary. Second, compare to the ILP based systems that optimize a heuristic objective over the entire summary (Gillick and Favre, 2009; Li et al., 2015), we use the training criteria (ROUGE scores) as the objective. Third, compare to other global inference methods that also optimize ROUGE scores, we use a rich set of features that encode the quality of the entire summary, while most prior methods utilize features that are defined on the sentence level (Aker et al., 2010; Kulesza and Taskar, 2012). Fourth, selecting among K-best summaries is relatively new in summarization. To the best of my knowledge, only very recent work done in parallel studies this problem independently (Wan et al., 2015).[1]

I have conducted a preliminary experiment on how to extend the combination pipeline for summarization. For the first step, we generate N word summaries using the ICSISumm model ($N = 250$). After that, we generate candidate summaries of 100 words by enumerating the sets of sentences from the longer summary (Section 6.4.2 describes how to do this). On average 452 summaries are generated per input. For the second step, we use the model described in Section 6.5. We exclude features based on consensus between the basic systems and system identity features because they are unavailable. The experiment setting is the same as what we have described in Section 6.7.1.

As can be seen in Table 8.1, the best possible performance (row Highest) among

[1] Wan et al. (2015) explore this idea based on a different implementation. In the first step, they also use Integer Linear Programming to generate K-best summaries, but based on a different objective. In the second step, they use SVMRank based on different features. Their method achieves 0.004-0.0045 improvement over their baseline in terms of R-2 on the DUC 02, 04 data.

these candidates is much higher than ICSISumm (i.e., the 1-best summary). However, our current model only achieves an improvement over the 1-best summary on the training/development set (DUC 03, 04) and the DUC 02 data. In my opinion, the main problem lies in the decoding stage, where the majority of candidate summaries have a low performance (as reflected in the median performance of the candidate summaries). Future work may investigate how this problem can be tackled.

	DUC 2001		DUC 2002		DUC 2003	
	R-1	R-2	R-1	R-2	R-1	R-2
ICSISumm	0.342	0.079	0.373	0.095	0.381	0.103
Our model	0.343	0.076	0.379	0.097	0.391	0.103
Highest	0.396	0.116	0.433	0.133	0.434	0.134
Median	0.330	0.067	0.352	0.077	0.362	0.082

	DUC 2004		TAC 2008		TAC 2009	
	R-1	R-2	R-1	R-2	R-1	R-2
ICSISumm	0.384	0.098	0.388	0.119	0.393	0.121
Our model	0.398	0.103	0.385	0.110	0.391	0.111
Highest	0.433	0.127	0.434	0.144	0.440	0.147
Median	0.362	0.082	0.366	0.091	0.375	0.095

Table 8.1: The performance of our baseline method (ICSISumm) and the current model. We also show macro average of the highest and median performance among the candidate summaries of an input.

Evaluation of Summarization Systems

In Section 4.2, we show that summaries from the state-of-the-art systems have similar performance. This means, the non-overlapping concepts from summaries of different systems are judged as equally good by current evaluation methods. However, this does not necessarily mean that humans do not have a preference towards a summary. Rather, it might be because these evaluation methods are too coarse. For example, ROUGE-n evaluates the recall of n-grams in machine summaries compared to human summaries. Apparently, paraphrases of the same content will be regarded as different by ROUGE-n. For future work, new methods can be developed to evaluate the recall of semantic content units automatically. Hovy et al. (2006) have explored this idea, where they propose to compute the recall of basic elements (BE) (e.g., "*the head of a major syntactic constituent, a relation between a head-BE and a single-dependent, expressed in a triple*" (Hovy et al., 2006)) in machine summaries. Though this method does not outperform ROUGE-n in terms of correlation with manual evaluation (Rankel et al., 2013), it would be interesting to continue exploring on this direction.

In summary, it is time to design more sensitive evaluation methods that are capable to capture finer nuances between systems. In natural language processing, progress in evaluation techniques always leads to progress in generation techniques. Our model in Chapter 6 directly optimizes the ROUGE score of the entire summary. A better evaluation method is developed is likely to benefit this model.

Appendix A

Sample Input Documents

Three input documents for the topic-based summarization problem in Table 1.1

AFP_ENG_20050119.0019

Headline: Indian, Pakistan military to discuss alleged Kashmir ceasefire violation

Dateline: NEW DELHI, Jan 19

Indian and Pakistani military commanders were to discuss Wednesday Indian charges that Pakistan fired mortar shells across the border into Indian-controlled Kashmir in violation of a 14-month ceasefire.

The director-generals of military operations of the nuclear-armed neighbours were slated to talk by telephone about the incident that occurred late Tuesday.

"The director generals of military operations will be talking later today," an Indian government official said, declining to be named. "The Pakistani side has denied firing. Let's see."

The time of the hotline call between the Indian and Pakistani officials was not immediately known.

On Wednesday, Islamabad denied it had violated the ceasefire in the disputed Himalayan state of Kashmir, spark of two of three wars between Indian and Pakistan, and over which they skirmished in 2002.

India and Pakistan began the ceasefire November 25, 2003, after routinely exchanging artillery fire across the volatile Line of Control, the de facto border separating their armies in Kashmir, as part of a tentative peace process.

"No one from Pakistan has fired and there is no ceasefire violation by Pakistan," Pakistan military spokesman Major General Shaukat Sultan told AFP in Islamabad.

The firing was front-page news in Indian newspapers. "Pakistan opens fire across Line of Control," said The Hindu.

Indian police said Tuesday at least one person was injured when about 15 mortars were fired into India from Pakistan over the Line of Control.

The mortars came from across the border Tuesday evening and landed in India's Durga Post area in the Poonch sector, an Indian police spokesman said.

Indian army officer Major General D. Samanwar told NDTV news channel the incident was tantamount to a ceasefire violation.

"Yes, it certainly is a violation. It's the first time it has happened but we've exercised full restraint," he said, adding troops had been put on alert.

The two sides have been engaged in formal peace talks to normalise relations since January 2004.

AFP_ENG_20050312.0019

Headline: Indian PM Singh to flag off first trans-Kashmir bus
Dateline: NEW DELHI, March 12

India's prime minister will personally flag off the first bus to travel between the Indian and Pakistan zones of divided Kashmir in almost six decades when the service resumes next month, an official said Saturday.

Prime Minister Manmohan Singh's visit to Kashmir in April will be his third since he assumed office in May.

"The prime minister will be in (the Indian Kashmir summer capital) Srinagar to flag off the first bus on April 7," Singh's spokesman said.

The bus service is the first tangible result of 14 months of dialogue between the nuclear-armed neighbours who have fought three wars including two over Kashmir, which each hold in part but claim in full.

India and Pakistan agreed last month that Kashmiri residents would not need passports to cross the divided state by bus but would use permits issued by the civil administration after being cleared by police.

The bus service was suspended in 1947 after the first India-Pakistan war over Kashmir. The second was in 1965.

Indian Kashmir is in the grip of a 15-year-old insurgency that has so far left thousands dead.

India blames Pakistan for fomenting the insurgency in Kashmir which has left at least 40,000 dead according Indian estimates, though separatists put the toll at twice that number.

Pakistan denies supporting the rebels though it admits to extending moral, political and diplomatic support to the Kashmiris' freedom struggle.

Indian authorities last week said some 150 permits had been issued to applicants from Indian-Kashmir to travel by the bus linking Srinagar to Muzaffarabad, capital of the Pakistan-administered zone.

Of the 150 applicants, 60 will be short-listed for the first two trips – the first on April 7 and the second a fortnight later – an official said.

APW_ENG_20041118.0081

Headline: Kashmiris divided by Pakistan-India conflict greet each other across river

Crying, waving and throwing letters wrapped around stones to each other, hundreds of Kashmiris gathered on either side of a river that separates the disputed Himalayan region between Pakistan and India.

The emotional reunion Wednesday across the Neelum River came on the final day of the Islamic holiday of Eid al-Fitr and as Indian Prime Minister Manmohan Singh paid a rare visit to Indian-controlled Kashmir, his country's only Muslim-majority state.

Families separated for decades by the Kashmir conflict shouted greetings over the noise of the fast-flowing river. Despite easing tensions between South Asia's nuclear rivals, Kashmiris are still forbidden from crossing the watery border.

Nosheen Akhtar, 22, and her mother, Khonshad Begum, 40, burst into tears, wailed and beat their chests unable to meet Begum's mother on the Indian side.

"I want to jump into the river. My mother is there and I cannot go over to meet with her," said Begum, her relatives holding her back.

Authorities on both sides were restricting access to the riverside at Karin, 90 kilometers (55 miles) north of Muzaffarabad, the capital of Pakistan-held Kashmir. Indian soldiers were preventing people from getting too close, and on the Pakistan side, police said authorities had not given formal permission for the reunions.

The Pakistani and Indian portions of Kashmir are separated by a heavily militarized cease-fire line where skirmishes were once common between their forces. The guns have fallen silent for the past year amid the thaw in relations, and since then there have been several opportunities for family reunions albeit at a distance of 25 meters (yards) across the river.

Pakistan and India both claim Kashmir in its entirety and have fought two wars over it since their independence from British rule in 1947. The conflict has only deepened since an Islamic insurgency began on the Indian side in 1989. India accuses Pakistan of helping foment it, which Pakistan denies. Some Kashmiris have fled from the Indian to the Pakistan side, accused of involvement with the rebels.

On Wednesday, Singh, the Indian leader, expressed a commitment to make peace with Pakistan, while Islamabad hailed India's withdrawal of some of India's troops in Kashmir a move hoped to spur the peace process.

But Kashmiris are impatient for progress that will allow them to reunite with relatives across the cease-fire line, known as the Line of Control.

The two countries are due to start formal talks on Kashmir next month, and people on both sides are eager for a long-standing proposal for a bus service between Muzaffarabad and Srinagar, the summer capital of Indian-held Kashmir, to finally get off the ground.

"Both the countries have not been allowing us to meet," said Raja Azhar Khan, a 55-year-old refugee, whose wife and son live on the Indian side of the Neelum River.

"The two countries are getting friendly. They are allowing people to meet each other at Wagha (the main land border between Indian and Pakistan). Why are we Kashmiris not being allowed to meet?"

Three input documents for the generic summarization problem in Table 2.1

APW19981018.0836

In a critical ruling for the North American National Basketball Association and the players' union, arbitrator John Feerick decides Monday whether more than 200 players with guaranteed contracts should be paid during the lockout. If the players win, the owners will be liable for about dlrs 800 million in guaranteed salaries, although they have vowed to appeal if they lose. The league already has sued the players over Feerick's jurisdiction. "If we win, I think it just emboldens the spirit and resolve of the players," union director Billy Hunter said. "But I don't think there will be anybody celebrating because there's no guarantee that it will end the lockout. "It only means they have to pay some 200 players, and they've indicated to us their intent to file an immediate appeal and take it as far as they have to in order to avoid payment. "So even if he does rule in our favor, at most it's a hollow victory. The players aren't going to get paid Nov. 15 in any circumstance," Hunter said. If the owners win, it will remove the last wild card the players had been holding. The sides have not negotiated since last Tuesday, when the union proposed a superstar tax on the highest contracts. The league made a counterproposal Friday, asking that the tax be imposed with a much lower threshold. Hunter dismissed the league's latest proposal on Friday afternoon, then said both sides would be best served by awaiting Feerick's ruling. It's unlikely any negotiations will be held this week, since the union is holding a meeting for all NBA players and the agents advisory committee in Las Vegas from Wednesday through Friday. "We've got to get a sense of where the players are, what they consider to be reasonable and what they're willing to do in order to get the season to commence," Hunter said. The union filed a grievance with Feerick before the lockout was imposed July 1 over the owners' announcement June 29 that they would not honor guaranteed deals. In a six-day hearing over the summer, the union argued that owners should have protected themselves from being liable for guaranteed salaries during a work stoppage by inserting lockout language into the standard player contract. The Sacramento Kings inserted a lockout clause into center Olden Polynice's contract in 1994, and it was approved by the league. The union used the existence of that clause to argue that all the other teams should have protected themselves similarly. Most players are due to receive their first paychecks Nov. 15, although a dozen or so had clauses entitling them to be paid over the summer. None has received a paycheck. The NBA argued that a tenet of labor law allows employers to withhold pay from employees during a lockout. The league also called former union director Simon Gourdine to testify. He said it was his understanding when he negotiated the old labor agreement in 1995 that players would not be paid if the owners chose to reopen the agreement and impose a lockout.

APW19981003.0083

He was the classic small-town prodigy, with the creativity of a big-city profiteer. When there was no shot to take, he invented a new one. When there was no one to pass to, he reconfigured the play until a teammate was open. Larry Bird, in the Indiana countryside or inside Boston Garden, was a luminous exception to the governing rule. That is why, six years after his retirement from the National Basketball Association, his name is again basketball's most prominent, beginning with his induction to the Hall of Fame before 7,000 Bird watchers at the Civic Center here on Friday night. Deservedly enshrined as forever exceptional, he again becomes Bird, the exception, the case study for a contentious and potentially disastrous labor war. "No, not really," Bird said, when I asked whether he is troubled by the likelihood of his legendary name soon representing a symbol of greed to unsympathetic millions. "There's always a player's name attached to these things. I know at the time I was very happy about it." That would have been 1988, when the Celtics wanted to compensate Bird with a $4.9 million bonus to push through his back pain, go on as their savior. Three years later, in a contract arbitration involving the Knicks' Patrick Ewing, the agent David Falk would contend that the NBA conspired with the Celtics to circumvent the salary cap, in order to satisfy Bird. Alan Greenspan, I am sure, would agree that this salary cap is convoluted enough to give anyone a headache, so let's just say it is a cap that does not exist when a team is negotiating with one of its own. The process of unsealing the cap to re-sign a particular player eventually became known as making use of the Larry Bird exception. And that is where we stand, as this onetime exception has become the very expensive rule the owners don't want to play by anymore. "I can understand both sides," said Bird, safely in the middle, between Bird, the former exception, and Bird, the present Indiana Pacers' coach. "Without getting into the exception, I think it's very important for players to stay in the same place." Important, he meant, for franchise stability and fan identification. "You have a son who is 7 years old, he goes from 7 to 17 in the 10 years you've played," Bird said. "A lot of people in Boston told me that they had followed me, from the time they were very young to when they were in college." The Bird years numbered only three Celtics championships, but he was the best player pro basketball's most famous team ever had. He and Magic Johnson created a basketball renaissance that began during a college title showdown in Salt Lake City and spread worldwide, like an infectious smile. They stood for the pass, for team play, but now their decade of selflessness has given way to one of selfishness. The NBA of Michael Jordan reached greater heights than anyone imagined it could, but it is a league that now suffers from a sickness of the soul. "If Larry and Magic hadn't done what they did, we might not survive what we're about to go through," said Bill Fitch, Bird's first Celtics coach, who, with Bill Walton, stood with him on the night that, he said, gave closure to his playing career. The owners, as always, are exaggerating their misery, but this time, it is much easier to not root for the players. The president of the union is Ewing, who one day commands players to boycott the world championships because the NBA's corporate fingerprints are on them, then the next day helps himself to some television commentary work for David Stern's women's annex. Ewing leads the fight to protect the $100 million contracts for 21-year-olds who have achieved not a single playoff victory, linked to the big payoffs for agents like his friend Falk. The battle is waged in the name of a salary cap that makes exceptions of the unexceptional, rewards everyone as if they were Bird. “I believe that in any field there has to be an allowance for the truly special ones,” Walton said. “But that group is very small. When I was growing up in this sport, the only players who got the recognition were the champions, the ones who always made you feel good about the game, about sports. That’s how Larry and Magic played, always dreaming of the special team. It wasn’t about hype, about money.” That is not quite the case, nor should it have been. Bird was a businessman’s ball player from the day he arrived, with his flannel shirts and blue-collar ethic. He hired the late Bob Woolf, one of the original heavy-hitting agents, and got himself a record rookie contract. Then he went out and turned a 29-victory catastrophe into a 61-victory contender. A rare Bird, an honest exception to the rule.

NYT19981008.0461

The first substantive talks in more than two months between opposing sides of the National Basketball Association’s labor dispute came and went Thursday without a hint of a settlement. Still, a five-hour meeting that was described as cordial by the league and “almost like two bulls letting off a little steam” by the players association produced another scheduled round of talks next Tuesday. Barring a major compromise, that will not be enough time to preserve a full season and prevent the league from losing its first regular-season games to labor strife in November. Russ Granik, the NBA’s deputy commissioner, said the league would wait until after next week’s meeting before deciding to cancel regular-season games. He also discussed the possibility of a significantly shortened season. “We haven’t made a determination that you need this exact number of games in order to have a representative season,” Granik said. “But we recognize that beyond a certain point we can’t possibly sell to our fans that we’re having an NBA season. “Whether’s that 60 games, 50 games or 49 or 53, we’re not there yet. We have a few months before we have to face that decision.” Perhaps the only progress involved Thursday in the conference room of a midtown Manhattan hotel was a question-and-answer session over the league’s latest proposal to the players. Patrick Ewing, the union president, and vice presidents Herb Williams and Dikembe Mutombo attended the meeting with the union’s executive director, Billy Hunter, and union lawyers. No owners were present, but Granik, Commissioner David Stern and the league’s lawyers spent most of the day explaining the intricacies of a two-week-old proposal to the players. At one point before the two parties broke for lunch, Stern and Hunter raised their voices and accused each other of handling their constituencies poorly, a participant in the meeting said. But the two were seen shaking hands and laughing after the meeting concluded shortly after 3 p.m. “There was some venting from both sides,” Hunter said. “We’ve been placid and very respectful. Today, we took the coats off and we were inclined to take the gloves off a little bit. Having done that, I think it kind of loosened up both sides.” Hunter added: “Did anybody blink today? They’re sort of look at us for any kind of nuance they can find during the course of negotiations that might, in some way or another, give some indication that while we’re mouthing one thing we might be open to something else. We’re looking at their body language, too. I don’t think that they’re ready to make a deal.” The last formal meeting between both sides on Aug. 6 ended when Stern and the owners abruptly marched out after they had received a proposal from the players.

Three input documents for the topic-based summarization problem in Table 2.1

AFP_ENG_20051015.0380

Headline: Planned US neo-Nazi march sparks rioting
Dateline: CHICAGO, Oct 15

Rioting erupted Saturday in an impoverished neighborhood of Toledo, Ohio after hundreds of counter-protestors broke up a planned march by a neo-Nazi group, Toledo police confirmed.

Scores of rioters rampaged through streets wrecking cars and etting a building on fire in north Toledo after police fired tear gas on the anti-Nazi group.

Six people were arrested, according to Fox News.

The violence broke out after a group of nearly 80 white supremacists of the Virigina-based National Socialist Movement assembled for a march against what they called black gang violence against whites in the Toledo neighborhood.

Bob White, leader of the movement, said police forced his group to disband after the protestors gathered around them and began to throw stones.

”The police lost total control of the situation,” he told AFP.

”We were protesting black racial violence against white people in that neighborhood,” said White.

But a local politician said that the multi-ethnic neighborhood was not particularly beset by violence.

"That neighborhood has never been known for racial tensions," city councilman Robert McCloskey told Fox.

APW_ENG_20050805.1073

Headline: Candidate drops out of Charlotte, North Carolina, council race after white supremacist writings exposed
Dateline: CHARLOTTE, North Carolina

A city council candidate dropped out of the race Friday after it was disclosed that he posted comments to a white supremacist Internet bulletin board more than 4,000 times.

Doug Hanks said the postings were fictional and designed to win white supremacists' trust as he researched a novel he was writing. He said the book was also meant to appeal to white supremacists.

"I needed information for the book and some other writings I was doing," Hanks told The Associated Press on Friday. "I did what I thought I needed to do to establish myself as a credible white nationalist."

Hanks had filed papers seeking the Republican nomination for one of four at-large council seats, but formally withdrew Friday, said county elections director Michael Dickerson.

Hanks' postings over the past three years were first reported by The Rhinoceros Times, a weekly newspaper. In one June 1 posting, he said blacks should be treated like "rabid beasts."

Hanks said his self-published novel, called "Patriot Act," takes themes from "The Turner Diaries"the racist novel believed to have inspired Oklahoma City bomber Timothy McVeigh.

"I saw how successful these 'Turner Diaries' had been and that was the path to take," he told the AP. White supremacists, he said, "have more of a tendency to word of mouth, to say, 'Hey, you ought to pick this up.'"

Republican Mayor Pat McCrory condemned Hanks.

"He's a man of total inner hatred in both his heart and soul, and it doesn't matter what party he's in," McCrory said.

APW_ENG_20050509.0484

Headline: Two arrested outside Boston Holocaust gathering; crowd protests white supremacists' presence
Dateline: BOSTON

White supremacists clashed with an angry crowd outside Faneuil Hall, where Holocaust survivors and their families were commemorating the liberation of Nazi concentration camps.

Two people were arrested during Sunday's confrontation, officer John Boyle said. Sunday was the 60th anniversary of the end of the Allied victory over Nazi Germany.

Inside the historic meeting house, Holocaust survivors, their children and grandchildren lit white candles to commemorate the estimated 6 million Jews killed by the Nazis. Germany's consul-general to New England, Wolfgang Vorwerk, spoke of his country's role.

Outside, 10 to 15 members of the Arkansas-based group White Revolution were escorted by officers to a designated protest area across the street. The officers, many in riot gear, formed a barricade between the protesters and about 100 people who angrily shouted at them to leave Boston.

Two people were charged with disturbing the peace after a scuffle outside the protest area. Police said Shireen Chambers, 36, of Boston, who is white, struck a black man, Jerome Higins, 25, of Everett, and he retaliated by spitting in her face and hitting her with a sign. Boyle said at least one officer was injured while arresting Chambers, who yelled racist epithets as she was dragged away.

Gov. Mitt Romney, who attended the Faneuil Hall event, said he was disgusted by the presence of the supremacists.

"Today of all days, to have white supremacists come here from Arkansas, is most disappointing," he said. "I wish they'd go back home where the came from and bury themselves under the rocks that they crawled out from."

Three documents for the Input A in Table 3.2 (d30020t in DUC 2003)

APW19981103.0271

Headline: North Korea to send 317-member delegation to Asian Games

SEOUL, South Korea (AP) _ Despite catastrophic hunger at home, North Korea plans to send 317 athletes and officials to next month's Asian Games in Thailand, South Korean officials said Thursday.

It will be the largest sports delegation the communist country has sent abroad in recent years.

North Korean Sports Minister Chang Ung said 209 athletes from his country will compete in 21 events in Bangkok, hoping to win medals in women's judo, women's soccer, wrestling, table tennis, weightlifting and boxing.

Chang made the remarks in an interview published recently by the Chosun Shinbo, a newspaper run by pro-North Korean residents in Japan, said Seoul's Naewoe Press, which obtained the report.

Chang said North Korea was sending a large delegation to Bangkok to prepare for the 2000 Sydney Olympics.

North Korea did not enter the last Asian Games, in Hiroshima, Japan, in 1994. It sent only 18 athletes and officials to the Winter Olympics in Nagano, Japan, in February.

Naewoe is run by South Korea's main government intelligence agency and specializes in monitoring communist news media.

Three years of floods and drought that started in 1995 devastated North Korea's collective farming and planned economy, forcing the country to rely on outside aid to feed its 23 million people.

APW19981206.0169

Headline: Former drug cheat back in the swim of things

BANGKOK, Thailand (AP) _ China's swim team will hit the water at the Asian Games on Monday with a reminder of the darkest chapter in its story of drug shame.

Xiong Gouming, one of the 11 Chinese athletes who tested positive for steroids in Hiroshima four years ago, will compete in the men's 400-meter individual medley, taking his place on a team that is constantly forced to defend itself.

Head coach Zhang Xiong said Xiong Gouming "feels fine" about being back at an Asian Games. "That was the past," said Zhang, brushing off drug questions Sunday. "We hope that we will be clean."

The Chinese Olympic Committee tested athletes before their departure for Bangkok, and Chinese media said the swimmers were subjected to four surprise tests conducted by world governing body FINA.

"We treat this as a major issue to make sure no Chinese athletes will test positive," COC vice president Li Furong said.

Earlier this year China announced, through FINA, that any of its athletes testing positive for steroids would be banned for life. Under those criteria, Xiong would never have had his second chance at an Asian Games.

Xiong won four golds in Hiroshima but has been mediocre since returning to international competition at the East Asian Games a year ago. He was one of many poor performers at the world championships in Perth in January, as the Chinese appeared demoralized by yet another scandal.

After female swimmer Yuan Yuan was caught trying to smuggle 13 vials of human growth hormone through Sydney Airport, four of her teammates were nabbed with masking agents in their systems and thrown out of the games.

Chinese women swimmers won 12 of the 16 golds at stake in the 1994 Rome world championships and all 15 of their races in the 1994 Asian Games.

The team won three gold medals at this year's world championships. Two of those went in the medley events to world record holders Chen Yan and Wu Yanyan, who will lead the Chinese team here.

Fellow world record holders Le Jingyi and He Cihong were left off a generally young squad assembled with a view ahead to the 2000 Olympics.

Japan's rising talent will test the inexperienced Chinese after a successful world championships.

Mai Nakamura, a silver medalist in the 100-meter backstroke and bronze medalist over 200; Ayari Aoyama, silver medalist in the 100-meter butterfly, and Yasuko Tajima, bronze medalist in the 400-meter medley, give the Japanese team undoubted class.

APW19981202.0568

Headline: Hat trick for China's Chen; South Korea beaten; Emirates win

BANGKOK, Thailand (AP) _ Star striker Chen Yang scored a hat trick Wednesday for China in a lackluster 4-1 victory over Cambodia, securing his country a spot in the second soccer round for the 13th Asian Games.

Underdog Turkmenistan also earned a second-round berth by overcoming a 2-0 half-time deficit to defeat South Korea in a nail-biting finish, with two goals coming in the final 10 minutes.

The United Arab Emirates and North Korea ended regulation play and extra time at 3-3, settling their Group E match in the southern city of Songkhla with a penalty shootout. The Emirates won, 7-4.

Playing at the National Stadium on a poor pitch with traces of day-old repairs visible, host Thailand made a fiery debut against Hong Kong, shutting out their opponents 5-0 after leading 3-0 at the half.

Thailand's Kritsada Piandit scored the first goal barely a minute into the match, followed by teammates Worawood Srimaka on a penalty in the eighth minute and Kiatisuk Senamuang in the 22nd minute.

Eight minutes into the second half, Tawan Sripan kicked in Thailand's fourth goal, and Kiatisuk hit the target for the second time at the 40th minute to make the final score 5-0.

The result takes Hong Kong out of competition, since they also lost their Group F opener to Oman on Monday.

China fulfilled expectations of topping Group B in Surat Thani with the victory over Cambodia, the 4-1 scoreline mirroring Monday's rout of Lebanon. But though the Chinese dominated, they failed to sparkle.

Chen scored early in the first half, heading in a long pass from midfield. Peng Wang caught a pass on the right side about 10 minutes later and fired his second Asian Games goal.

Chen scored again on a near shot and headed in his third goal in the 38th minute.

Cambodia partially salvaged its honor late in the second half when Hok Sochetra beat the complacent Chinese defense and scored.

It was one of the only shots on goal that the Cambodians had.

Following Monday's victory over regional favorite Vietnam, Turkmenistan staged its second upset in Group A in the central city of Nakhon Sawan, eclipsing South Korea.

The South Koreans carved up the Turkmenistan defense in the first half. Yong-Soo Choi gave his team which hoped to erase memories of a dismal World Cup a commanding two-goal advantage with an opening-minute goal and a shot that pin-balled off a teammate into the net in the 44th minute.

Turkmenistan regrouped after the half. A long cross found the head of Igor Kislov to put Turkmenistan on the scoreboard. Steady pressure led to a dangerous foul and red card for South Korea's Byoung-Keun Lee.

Muslim Agaev took advantage of the reduced South Korean side with a slicing left-footed shot into the top far corner to equalize. Kislov fired home the winner in the dying minutes.

Three documents for the Input B in Table 3.2 (d31050t in DUC 2003)

APW19981202.1274

Headline: China holds high-profile dissidents, colleagues demand release

BEIJING (AP) _ China's central government ordered the arrest of a prominent democracy campaigner and may use his contacts with exiled Chinese dissidents to charge him with harming national security, a colleague said Wednesday.

Two Beijing police officers spent 30 minutes telling Zha Jianguo to stop trying to set up a political opposition party. Underscoring the warning, they said his colleague, Xu Wenli, won't be released soon and may be charged for having links to "reactionary groups," Zha said.

Xu and another influential dissident, Qin Yongmin, were arrested Monday night in police raids in two cities that delivered the sternest blow so far to a five-month campaign to establish the China Democracy Party and challenge the ruling Communist Party's monopoly on power.

Qin was arrested for plotting to overthrow the government, a crime that could land him in jail for life. A third Democracy Party advocate, Wang Youcai, already in custody for a month, was also formally arrested Monday although his family has not been informed of the charges.

Zha, who helped Xu organize would-be party members in Beijing and the nearby port of Tianjin, said police officers told him Xu's arrest was ordered by the central government, not Beijing police.

He took the police reference to "reactionary groups" to mean exiled dissidents in the United States. Under China's vague State Security Law, such links may also be punishable by up to life in prison.

Zha pledged to work with dissidents in China and exiles in the United States to campaign "to save Xu Wenli."

On Wednesday, 190 dissidents from around the country demanded in an open letter that the government release Xu, Qin and Wang Youcai, saying the arrests run counter to U.N. human rights treaties China has signed over the past 14 months.

The authorities "are deceiving and cheating international public opinion while on the other hand they are suppressing and persecuting domestic political dissidents," said the letter faxed to foreign news agencies.

In Washington, White House spokesman Joe Lockhart said the United States deplored the detention and arrests of Xu and Qin.

"We believe the peaceful political activities of this kind and other forms of peaceful expression that they've been involved in are fundamental human rights that should be protected by all governments," Lockart said. "We call on the Chinese government to assure the protection in these cases of Mr. Xu and Mr. Qin."

State Department spokesman James P. Rubin said U.S. officials conveyed their concerns to the Chinese Ministry of Foreign Affairs and urged that Xu be released immediately.

U.S. officials received confirmation Wednesday that Xu is being detained on suspicion of "having conducted activities damaging to China's national security," Rubin said.

He said he had no information from Chinese authorities about Qin or Wang.

Two other democracy party supporters taken into custody in central Wuhan city along with Qin Chen Zhonghe and Xiao Shichang were released Wednesday morning, said He Xintong, Xu Wenli's wife. She added that police questioned the pair about the party as well as Qin's human rights monitoring organizations.

Qin and Xu are towering figures in China's persecuted dissident community. Their activism dates to the seminal Democracy Wall movement of the 1970s. Wang was a student leader in 1989's influential Tiananmen Square democracy movement. All have spent time in prison, Xu for 12 years, much of it in solitary confinement.

Xu's wife said she does not know where he is being held and, in her 20-year experience with the authorities, believes they are unlikely to tell her.

Released in 1993, Xu picked up his campaigning for political change soon after his parole ended last year. He has tried to use China's nascent legal system and the international treaties it signed to push for reform.

"My husband is innocent and there's nothing he can be criticized for," said his wife, He Xintong. "They're going to have to expend a lot of effort to make him a criminal."

NYT19981209.0542

Headline: DISSIDENT FROM SHANGHAI ARRIVES IN NEW YORK

NEW YORK _ A Chinese dissident fleeing a new round of arrests of democracy activists in Shanghai arrived here Wednesday and announced that he and other opponents of the Chinese government plan a demonstration Thursday at the United Nations to protest the crackdown.

The dissident, Yao Zhenxian, who was released in April from a Chinese labor camp, is a leader of the China Democracy Party, which was formed in June during President Clinton's visit to China.

Speaking through an interpreter at Kennedy International Airport, Yao, 44, said little about why he had left Shanghai, except that he and his younger brother, Yao Zhenxiang, 38, had been sent to a labor camp in 1996 on a "trumped-up charge" of publishing pornography.

The younger Yao, who is also a prominent figure in the China Democracy Party, is scheduled for release in April.

Last week the Chinese government arrested 10 members and sympathizers of the China Democracy Party, one of whom, Wang Youcai, is to go on trial Dec. 17.

"The Chinese government feels it expanded too quickly," Yao said at the airport, referring to his party. Washington gave Yao a special visa, which expires in February.

Dr. Wang Bingzhang, 50, an adviser to the overseas committee of the party, said later that Yao had left China for personal and political reasons.

"They were scared all the time," he said, referring to Yao's wife, Yu Yingzhang, and daughter, Yao Yiting, 14, who accompanied him.

"The family had a terrible life, especially the daughter. Secret agents followed them all the time."

The family trading business, which was 12 years old, was shuttered by the government in 1996, said Wang, a former surgeon in Beijing, "so they had no way of living."

In addition, after Yao made organizational trips to several provinces, democracy activists urged him to go abroad, Wang said.

"They wanted him to tell the truth about what is really happening in China and to call on the whole world to pay attention," Wang said.

Beatrice Laroche, liaison at the United Nations for Human Rights in China, a New York-based group, said the China Democracy Party was a growing presence in some of China's most populous provinces, including Sichuan.

"But their most vocal leaders have all recently been detained," she added.

Ms. Laroche said the Yao brothers, especially the younger man, won prominence by helping to finance predecessors of the China Democracy Party with money from the family business before it was closed by the government.

NYT19981202.0309

Headline: CHINA DEFENDS ARREST OF PROMINENT DISSIDENT

BEIJING _ In response to criticism from home and abroad, Chinese officials broke their silence Wednesday to defend their arrest this week of a prominent dissident who was trying to form an opposition political party.

“Xu Wenli is suspected of involvement in activities damaging to national security and has violated relevant criminal codes of the People’s Republic of China,” said a statement from the Foreign Ministry, which on Tuesday declined to comment on the arrest.

The sudden arrest on Monday night of Xu, as well as several other activists involved with him in trying to form the China Democratic Party, set off strong protests from human rights groups, other Chinese dissidents and Washington.

“We view his detention for peacefully exercising fundamental freedoms guaranteed by international human rights instruments as a serious step in the wrong direction,” State Department spokesman James P. Rubin said in Washington on Tuesday. U.S. officials in Beijing urged the government to release Xu and asked for clarification as to the exact nature of his crime.

China signed the International Covenant of Civil and Political Rights with great fanfare in October, and Xu’s arrest is seen by human rights groups as a test of the nation’s commitment to its tenets.

Dissidents in and out of China rose to Xu’s defense, with more than a dozen activists around the country announcing that they would begin fasts in support of Xu and another leader of the China Democratic Party, Qin Yongmin, who was arrested in his home in Wuhan on Monday.

Almost 200 dissidents signed a letter to the Chinese government protesting the detentions, said the Information Center for Human Rights and Democratic Movement in Hong Kong.

Three other Democratic Party organizers were also detained on Monday, although two of them were released early Wednesday.

But the two more prominent dissidents, Xu and Qin, are likely to face a much longer haul since both have been charged with “criminal acts.”

Xu’s wife, He Xintong, said Wednesday night that she had still not been informed of the specific charge against her husband, although she surmised from the aggressive behavior of the arresting officers that the sentence “could be long.”

Qin’s family was told that he was charged with “plotting to subvert the government,” a crime that for serious offenses commands sentences of three years to life.

(STORY CAN END HERE. OPTIONAL MATERIAL FOLLOWS)

In the Chinese criminal code, this charge comes under a grab-bag section called “threatening state security,” which makes almost any political activity that questions or hampers the authority of the Communist Party illegal, from “violent or nonviolent activities aimed at overthrowing the government authorities,” to “activities designed to change the basic nature of the state.”

Xu’s and Qin’s trouble almost certainly stems from their efforts to gain recognition for the China Democratic Party, a loose network of pro-democracy activists in more than a dozen cities around China that was formed this year.

In the last six months they have become increasingly aggressive and defiant in their attempts to register the party with the government, submitting repeated applications even after local authorities had declared the concept of an opposition party illegal. They say the Chinese Constitution does not specifically forbid the formation of new political parties, although there have been no new parties since the founding of the People’s Republic in 1949.

In fact, the by-laws of the China Democratic Party are fairly tame; they carefully acknowledge the central role of the Communist Party, but also support free speech and free elections for public officials.

“My husband thought the time was right to begin working to form a new party, since China recently signed the covenant on human rights,” Ms. He said. In September, a few Democratic Party members got some slightly encouraging signals from local governments, which initially accepted their applications to form a social organization to develop a party. But in recent weeks, as organizers like Xu became more insistent and defiant, harassment by the police increased.

“All this past week we felt something was going to happen,” Ms. He said. “It seemed that anyone who came to visit us was later detained for a while. And there have been a lot more cars from the Public Security Bureau parked outside than is usual.”

Appendix B

A Dictionary of Words and their Abbreviations for the New York Times dataset

Below we show a hand-crafted dictionary that includes words and their typical abbreviations used during analysis for the NYT corpus. Words in the same line are regarded as the same, split by "—". During the preprocessing stage introduced in Method 2 of Section 5.4.4, a word in a summary or article that has appeared in the dictionary is replaced by its rightmost word in the dictionary.

pres—president
sen—sen.—senator
ny—ny.—new york
min—minister
nj—nj.—n.j.—new jersey
nyc—nyc.—new york city
dept—dept.—department
corp—corp.—corporation

co—co.—company

sec—sec.—secretary

gen—gen.—general

rep—rep.—representative

prof—prof—professor

un—u.n.—united nations

calif—calif.—california

conn—connecticut

cia—c.i.a.

fbi—f.b.i.

fla—fla.

dc—d.c.

sens—senators

fda—f.d.a.

ibm—i.b.m.

exec—executive

amb—ambassador

ga—ga.—georgia

gm—g.m.

asst—assistant

dist—district

atty—attorney—attorneys

mr—mr.

dr—dr.

ms—ms.

mrs—mrs.

jan—jan.—january

feb—feb.—february

mar—mar.—march

apr—apr.—april

may—may.—may

jun—jun.—june

jul—jul.—july

aug—aug.—august

sept—sep.—september

oct—oct.—october

nov—nov.—november

dec—dec.—december

gov—gov.—government

nfl—n.f.l.

nba—n.b.a.

mass—mass.—massachusetts

inc—inc.—incoporation

rev—rev.—revenue

ncaa—n.c.a.a.

va—va.—virginia

lt—lt.

pa—pa.—pennyslvania

tex—tex.

suv—s.u.v.

mich—mich.—michigan

irs—i.r.s.

md—md.

maj—maj.

aig—a.i.g.

col—col.—column

nhl—n.h.l.

ill—ill.—illinois

colo—colo.—colorado

st—st.

am—a.m.

pm—p.m.

jr—jr.

sr—sr.

us—u.s.—usa

ok—o.k.

va—va.—virginia

b—b.

c—c.

d—d.

e—e.

f—f.

g—g.

h—h.

j—j.

k—k.

l—l.

m—m.

n—n.

p—p.

q—q.

r—r.

s—s.

t—t.

u—u.

v—v.

w—w.

x—x.

y—y.

z—z.

References

Ahmet Aker, Trevor Cohn, and Robert Gaizauskas. 2010. Multi-document summarization using A* search and discriminative learning. In *Proceedings of EMNLP*, pages 482–491.

Miguel Almeida and André F.T. Martins. 2013. Fast and robust compressive summarization with dual decomposition and multi-task learning. In *Proceedings of ACL*, pages 196–206.

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-document abstractive summarization using ILP based multi-sentence compression. In *Proceedings of IJCAI*.

Srinivas Bangalore, German Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *Proceedings of ASRU*, pages 351–354.

Beata Beigman Klebanov, Nitin Madnani, Jill Burstein, and Swapna Somasundaran. 2014. Content importance models for scoring writing from sources. In *Proceedings of ACL: Short Papers*, pages 247–252.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of ACL-HLT*, pages 481–490.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022.

Olivier Bodenreider. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(1):267–270.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015a. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of AAAI*.

Ziqiang Cao, Furu Wei, Sujian Li, Wenjie Li, Ming Zhou, and Houfeng Wang. 2015b. Learning summary prior representation for extractive summarization. In *Proceedings of ACL-IJCNLP: Short Papers*, pages 829–833.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR*, pages 335–336.

Asli Celikyilmaz and Dilek Hakkani-Tur. 2010. A hybrid hierarchical model for multi-document summarization. In *Proceedings of ACL*, pages 815–824.

Asli Celikyilmaz and Dilek Hakkani-Tür. 2011. Discovery of topically coherent sentences for extractive summarization. In *Proceedings of ACL-HLT*, pages 491–499.

Daniel Cer, Christopher D. Manning, and Dan Jurafsky. 2013. Positive diversity tuning for machine translation system combination. In *Proceedings of WMT*, pages 320–328.

Hakan Ceylan, Rada Mihalcea, Umut Özertem, Elena Lloret, and Manuel Palomar. 2010. Quantifying the limits and success of extractive summarization systems across domains. In *Proceedings of ACL*, pages 903–911.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of ACL*, pages 173–180.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2013. Towards coherent multi-document summarization. In *Proceedings of HLT-NAACL*, pages 1163–1173.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

John M. Conroy, Jade Goldstein, Judith D. Schlesinger, and Dianne P. O'Leary. 2004. Left-brain/right-brain multi-document summarization. In *Proceedings of DUC*.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. 2006a. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of COLING/ACL*, pages 152–159.

John M. Conroy, Judith D. Schlesinger, Dianne P. O'Leary, and Jade Goldstein. 2006b. Back to basics: CLASSY 2006. In *Proceedings of DUC*.

John M. Conroy, Judith D. Schlesinger, Jeff Kubina, Peter A. Rankel, and Dianne P. O'Leary. 2011. CLASSY 2011 at TAC: Guided and multi-lingual summaries and evaluation metrics. In *Proceedings of TAC*.

John M. Conroy, Sashka T. Davis, Jeff Kubina, Yi-Kai Liu, Dianne P. O'Leary, and Judith D. Schlesinger. 2013. Multilingual summarization: Dimensionality reduction and a step towards optimal term coverage. In *Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization*, pages 55–63.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 update summarization task. In *Proceedings of TAC*, pages 1–16.

Dipanjan Das and André F.T. Martins. 2007. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU.*

Anirban Dasgupta, Ravi Kumar, and Sujith Ravi. 2013. Summarization through submodularity and dispersion. In *Proceedings of ACL*, pages 1014–1022.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of ACL*, pages 305–312.

Sashka T. Davis, John M. Conroy, and Judith D. Schlesinger. 2012. OCCAMS – An Optimal Combinatorial Covering Algorithm for Multi-document Summarization. In *ICDM Workshops*, pages 454–463.

Anthony Christopher Davison. 1997. *Bootstrap methods and their application*, volume 1. Cambridge university press.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, pages 449–454.

Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the Society for Information Sciences*, 41(6):391–407.

Jacob Devlin and Spyros Matsoukas. 2012. Trait-based hypothesis selection for machine translation. In *Proceedings of ACL-HLT: Short Papers*, pages 528–532.

Harris Drucker, Chris J.C. Burges, Linda Kaufman, Alex Smola, Vladimir Vapnik, et al. 1997. Support vector regression machines. In *Proceedings of NIPS*, pages 155–161.

Jesse Dunietz and Daniel Gillick. 2014. A new entity salience task with millions of training examples. In *Proceedings of EACL: Short Papers*, pages 205–209.

Daniel M. Dunlavy, John M. Conroy, Judith D. Schlesinger, Sarah A. Goodman, Mary Ellen Okurowski, Dianne P. O'Leary, and Hans van Halteren. 2003. Performance of a three-stage system for multi-document summarization. In *Proceedings of DUC*.

Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61–74.

Harold P. Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285.

Brigitte Endres-Niggemeyer and Elisabeth Neugebauer. 1998. Professional summarizing: no cognitive simulation without observation. *Journal of the American Society for Information Science*, 49(6):486–506.

Gunes Erkan and Dragomir R. Radev. 2004. LexRank: graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, pages 417–422.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. Event-based extractive summarization. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 104–111.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of ACL*, pages 363–370.

Jonathan G. Fiscus. 1997. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proceedings of ASRU*, pages 347–354.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and et al. 1999. Domain-specific keyphrase extraction. In *Proceedings of IJCAI*, pages 668–673.

Robert Frederking and Sergei Nirenburg. 1994. Three heads are better than one. In *Proceedings of ANLP*, pages 95–100.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing.*

Dan Gillick, Benoit Favre, and Dilek Hakkani-Tur. 2008. The ICSI summarization system at TAC 2008. In *Proceedings of the TAC.*

Dan Gillick, Benoit Favre, Dilek Hakkani-Tur, Berndt Bohnet, Yang Liu, and Shasha Xie. 2009. The ICSI/UTD Summarization System at TAC 2009. In *Proceedings of TAC.*

Daniel Jacob Gillick. 2011. *The elements of automatic summarization.* Ph.D. thesis, University of California, Berkeley.

Kevin Gimpel, Dhruv Batra, Chris Dyer, Gregory Shakhnarovich, and Virginia Tech. 2013. A systematic exploration of diversity in machine translation. In *Proceedings of EMNLP*, pages 1100–1111.

Yihong Gong and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Proceedings of SIGIR*, pages 19–25.

Yvette Graham. 2015. Re-evaluating Automatic Summarization with BLEU and 192 Shades of ROUGE. In *Proceedings of EMNLP*, pages 128–137.

Vishal Gupta and Gurpreet Singh Lehal. 2010. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence*, 2(3):258–268.

Surabhi Gupta, Ani Nenkova, and Dan Jurafsky. 2007. Measuring importance and query relevance in topic-focused multi-document summarization. In *Proceedings of ACL*, pages 193–196.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of HLT-NAACL*, pages 362–370.

Sanda Harabagiu and Finley Lacatusu. 2005. Topic themes for multi-document summarization. In *Proceedings of SIGIR 2005*, pages 202–209.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of ACL*, pages 1262–1273.

John C. Henderson and Eric Brill. 1999. Exploiting diversity for natural language processing: Combining parsers. In *Proceedings of EMNLP*, pages 187–194.

Kai Hong and Ani Nenkova. 2014a. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of EACL*, pages 712–721.

Kai Hong and Ani Nenkova. 2014b. Improving the estimation of word importance for news multi-document summarization—extended technical report. In *Technical Reports MS-CIS-14-02, University of Pennsylvania.*

Kai Hong, Christian G. Kohler, Mary E. March, Amber A. Parker, and Ani Nenkova. 2012. Lexical differences in autobiographical narratives from schizophrenic patients and healthy controls. In *Proceedings of EMNLP-CoNLL*, pages 37–47.

Kai Hong, John M. Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repositary of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of LREC*, pages 1608–1616.

Kai Hong, Mitchell Marcus, and Ani Nenkova. 2015a. System combination for multi-document summarization. In *Proceedings of EMNLP*, pages 107–117.

Kai Hong, Ani Nenkova, Mary E. March, Amber P. Parker, Ragini Verma, and Christian G. Kohler. 2015b. Lexical use in emotional autobiographical narratives of persons with schizophrenia and healthy controls. *Psychiatry research*, 225(1-2):40–49.

Eduard Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. 2006. Automated summarization evaluation with basic elements. In *Proceedings of LREC*, pages 604–611.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of EMNLP*, pages 216–223.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of KDD*, pages 133–142.

Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of KDD*, pages 217–226.

Samir Khuller, Anna Moss, and Joseph Naor. 1999. The budgeted maximum coverage problem. *Inf. Process. Lett.*, 70(1):39–45.

Jon M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632.

Alex Kulesza and Ben Taskar. 2012. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3).

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of SIGIR*, pages 68–73.

Wonkyum Lee, Jungsuk Kim, and Ian Lane. 2014. Multi-stream combination for LVCSR and keyword search on GPU-accelerated platforms. In *Proceedings of ICASSP*, pages 3296–3300.

Wenjie Li, Mingli Wu, Qin Lu, Wei Xu, and Chunfa Yuan. 2006. Extractive summarization using inter and intra event relevance. In *Proceedings of COLING-ACL*, pages 369–376.

Chen Li, Fei Liu, Fuliang Weng, and Yang Liu. 2013a. Document summarization via guided sentence compression. In *Proceedings of EMNLP*, pages 490–500.

Chen Li, Xian Qian, and Yang Liu. 2013b. Using supervised bigram-based ILP for extractive summarization. In *Proceedings of ACL*, pages 1004–1013.

Chen Li, Yang Liu, and Lin Zhao. 2015. Using external resources and joint learning for bigram weighting in ILP-based multi-document summarization. In *Proceedings of NAACL-HLT*, pages 778–787.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Proceedings of NAACL-HLT*, pages 912–920.

Hui Lin and Jeff Bilmes. 2011. A class of submodular functions for document summarization. In *Proceedings of ACL*, pages 510–520.

Hui Lin and Jeff Bilmes. 2012. Learning mixtures of submodular shells with application to document summarization. In *Proceedings of UAI*.

Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedgins of COLING*, pages 495–501.

Chin-Yew Lin and Eduard Hovy. 2003a. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of NAACL-HLT*, pages 71–78.

Chin-Yew Lin and Eduard Hovy. 2003b. The potential and limitations of automatic sentence extraction for summarization. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop*, pages 73–80.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81.

Marina Litvak and Mark Last. 2008. Graph-based keyword extraction for single-document summarization. In *Proceedings of COLING: workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 17–24.

Marina Litvak, Mark Last, and Menahem Friedman. 2010. A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of ACL*, pages 927–936.

Lemao Liu and Liang Huang. 2014. Search-aware tuning for machine translation. In *Proceedings of EMNLP*, pages 1942–1952.

Yu-Ting Liu, Tie-Yan Liu, Tao Qin, Zhi-Ming Ma, and Hang Li. 2007. Supervised rank aggregation. In *Proceedings of WWW*, pages 481–490.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of EMNLP*, pages 366–376.

Fei Liu, Feifan Liu, and Yang Liu. 2011. A supervised framework for keyword extraction from meeting transcripts. *Transactions on Audio Speech and Language Processing*, 19(3):538–548.

Patrice Lopez and Laurent Romary. 2010. HUMB: Automatic key term extraction from scientific articles in GROBID. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 248–251.

Annie Louis and Ani Nenkova. 2009. Automatically evaluating content selection in summarization without human models. In *Proceedings of EMNLP*, pages 306–314.

Annie Louis and Ani Nenkova. 2011a. Automatic identification of general and specific sentences by leveraging discourse annotations. In *Proceedings of IJCNLP*, pages 605–613.

Annie Louis and Ani Nenkova. 2011b. Text specificity and impact on quality of news summaries. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 34–42.

Annie Louis and Ani Nenkova. 2013. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2):267–300.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, April.

Wolfgang Macherey and Franz Josef Och. 2007. An empirical study on computing consensus translations from multiple machine translation systems. In *Proceedings of EMNLP-CoNLL*, pages 986–995.

Lidia Mangu, Eric Brill, and Andreas Stolcke. 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373–400.

Inderjeet Mani and Mark T. Maybury. 1999. *Advances in automatic text summarization*, volume 293. MIT Press.

Inderjeet Mani. 2001a. *Automatic summarization*, volume 3. John Benjamins Publishing.

Inderjeet Mani. 2001b. Summarization evaluation: An overview. In *Proceedings of NAACL 2001 Workshop on Automatic Summarization.*

Henry B. Mann and Donald R. Whitney. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60.

Sukanya Manna, Byron J. Gao, and Reed Coke. 2012. A subjective logic framework for multi-document summarization. In *Proceedings of COLING*, pages 797–808.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of ACL: System Demonstrations*, pages 55–60.

Rebecca Mason and Eugene Charniak. 2011. Extractive multi-document summaries should explicitly not contain document-specific content. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, pages 49–54.

Pawel Matykiewicz, Wlodzislaw Duch, and John P. Pestian. 2009. Clustering semantic spaces of suicide notes and newsgroups articles. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 179–184.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of ECIR*, pages 557–564.

Kathleen McKeown and Dragomir R. Radev. 1995. Generating summaries of multiple news articles. In *Proceedings of SIGIR*, pages 74–82.

Olena Medelyan, Eibe Frank, and Ian H. Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of EMNLP*, pages 1318–1327.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of EMNLP*, pages 404–411.

Rada Mihalcea. 2005. Language independent extractive summarization. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 49–52.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*.

Olof Mogren, Mikael Kågebäck, and Devdatt Dubhashi. 2015. Extractive summarization by aggregating multiple similarities. In *Proceedings of RANLP*.

Ahmed A. Mohamed and Sanguthevar Rajasekaran. 2005. A text summarizer based on meta-search. In *Proceedings of ISSPIT*, pages 670–674.

Saif Mohammad, Bonnie Dorr, Melissa Egan, Ahmed Hassan, Pradeep Muthukrishan, Vahed Qazvinian, Dragomir Radev, and David Zajic. 2009. Using citations to generate surveys of scientific paradigms. In *Proceedings of NAACL*, pages 584–592.

Hajime Morita, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2013. Subtree extractive summarization via submodular maximization. In *Proceedings of ACL*, pages 1023–1032.

Gabriel Murray, Steve Renals, Jean Carletta, and Johanna Moore. 2006. Incorporating speaker and discourse features into speech summarization. In *Proceedings of NAACL*, pages 367–374.

Ani Nenkova and Annie Louis. 2008. Can You Summarize This? Identifying Correlates of Input Difficulty for Multi-Document Summarization. In *Proceedings of ACL*, pages 825–833.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Foundations and Trends in Information Retrieval*, pages 103–233.

Ani Nenkova and Kathleen McKeown. 2012. A survey of text summarization techniques. *Mining Text Data*, pages 43–76.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proceedings of NAACL*, pages 145–152.

Ani Nenkova and Lucy Vanderwende. 2005. The impact of frequency on summarization. Technical report, Microsoft Research.

Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of SIGIR*, pages 573–580.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The Pyramid Method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4(2), May.

Ani Nenkova. 2005. Automatic text summarization of newswire: Lessons learned from the document understanding conference. In *Proceedings of AAAI*, pages 1436–1441.

Ani Nenkova. 2006. *Understanding the process of multi-document summarization: content selection, rewriting and evaluation.* Ph.D. thesis, Columbia University.

Matthew L. Newman, James W. Pennebaker, Diane S. Berry, and Jane M. Richards. 2003. Lying words: Predicting deception from linguistic styles. *Personality and social psychology bulletin*, 29(5):665–675.

Hermann Ney, Ute Essen, and Reinhard Kneser. 1994. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech & Language*, 8(1):1–38.

Andrew Y. Ng. 2004. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proceedings of ICML*.

Tadashi Nomoto. 2004. Multi-engine machine translation with voted language model. In *Proceedings of ACL*, pages 494–501.

Benjamin Nye and Ani Nenkova. 2015. Identification and characterization of newsworthy verbs in world news. In *NAACL-HLT: Short Papers*, pages 1440–1445.

Jahna Otterbacher, Günes Erkan, and Dragomir R. Radev. 2009. Biased LexRank: Passage retrieval using random walks with question-based priors. *Information Processing and Management*, 45(1):42–54.

You Ouyang, Wenjie Li, Sujian Li, and Qin Lu. 2011. Applying regression models to query-focused multi-document summarization. *Information Processing & Management*, 47(2):227–237.

Paul Over, Hoa Dang, and Donna Harman. 2007. DUC in context. *Inf. Process. Manage.*, 43(6):1506–1520.

Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An assessment of the accuracy of automatic evaluation in summarization. In *NAACL-HLT 2012: Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1998. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University.

Chris D. Paice and Paul A. Jones. 1993. The identification of important concepts in highly structured technical papers. In *Proceedings of SIGIR*, pages 69–78.

Chris D. Paice. 1980. The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In *Proceedings of SIGIR*, pages 172–191.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Michael Paul, Takao Doi, Young-Sook Hwang, Kenji Imamura, Hideo Okuma, and Eiichiro Sumita. 2005. Nobody is perfect: ATR's hybrid approach to spoken language translation. In *Proceedings of IWSLT*, pages 45–52.

Yulong Pei, Wenpeng Yin, Qifeng Fan, and Lian'en Huang. 2012. A supervised aggregation framework for multi-document summarization. In *Proceedings of COLING*, pages 2225–2242.

Emily Pitler, Annie Louis, and Ani Nenkova. 2010. Automatic evaluation of linguistic quality in multi-document summarization. In *Proceedings of ACL*, pages 544–554.

Joseph J. Pollock and Antonio Zamora. 1975. Automatic abstracting research at chemical abstracts service. *Journal of Chemical Information and Computer Sciences*, 15(4):226–232.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Vahed Qazvinian and Dragomir R. Radev. 2008. Scientific paper summarization using citation summary networks. In *Proceedings of COLING*, pages 689–696.

Dragomir R. Radev and Kathleen R. McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):470–500.

Dragomir R. Radev, Hongyan Jing, and Malgorzata Budzikowska. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization*, pages 21–30.

Dragomir R. Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, and Zhu Zhang. 2004a. MEAD-a platform for multidocument multilingual text summarization. In *Proceedings of LREC*, pages 1–4.

Dragomir R. Radev, Hongyan Jing, Malgorzata Styś, and Daniel Tam. 2004b. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

Peter A. Rankel, John M. Conroy, Eric Slud, and Dianne O'Leary. 2011. Ranking human and machine summarization systems. In *Proceedings of EMNLP*, pages 467–473.

Peter A. Rankel, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2013. A decade of automatic content evaluation of news summaries: Reassessing the state of the art. In *Proceedings of ACL: Short Papers*, pages 131–136.

Donna R. Recht and Lauren Leslie. 1988. Effect of prior knowledge on good and poor readers' memory of text. *Journal of Educational Psychology*, 80(1):16.

Antti-Veikko I. Rosti, Necip Fazil Ayan, Bing Xiang, Spyridon Matsoukas, Richard M. Schwartz, and Bonnie J. Dorr. 2007. Combining outputs from multiple machine translation systems. In *Proceedings of HLT-NAACL*, pages 228–235.

Rudolf J. Rummel. 1988. *Applied factor analysis.* Northwestern University Press.

Kenji Sagae and Alon Lavie. 2006. Parser combination by reparsing. In *Proceedings of NAACL: Short Papers*, pages 129–132.

Horacio Saggion, Juan-Manuel Torres-Moreno, Iria da Cunha, and Eric SanJuan. 2010. Multilingual summarization evaluation without human models. In *Proceedings of COLING*, pages 1059–1067.

Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley. 1997. Automatic text structuring and summarization. *Information Processing & Management*, 33(2):193–207.

Gerard Salton. 1971. *The SMART Retrieval System: Experiments in Automatic Document Processing.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Evan Sandhaus. 2008. The New York Times Annotated Corpus. *Linguistic Data Consortium, Philadelphia, PA.*

Beatrice Santorini. 1990. Part-of-speech tagging guidelines for the Penn Treebank Project. *Technical report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania.*

Barry Schiffman, Ani Nenkova, and Kathleen McKeown. 2002. Experiments in multidocument summarization. In *Proceedings of HLT*, pages 52–58.

Frank Schilder and Ravikumar Kondadadi. 2008. FastSum: fast and accurate query-based multi-document summarization. In *Proceedings of ACL-HLT: Short Papers*, pages 205–208.

Natalie Schluter and Anders Søgaard. 2015. Unsupervised extractive summarization via coverage maximization with syntactic and semantic concepts. In *Proceedings of ACL-IJCNLP: Short Papers*, pages 840–844.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proceedings of HLT-NAACL*, pages 177–184.

Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. 2012. Large-margin learning of submodular summarization models. In *Proceedings of EACL*, pages 224–233.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Josef Steinberger and Karel Jezek. 2004. Using latent semantic analysis in text summarization and summary evaluation. In *Proceedings of ISIM*, pages 93–100.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 2, pages 901–904.

Philip J. Stone, Dexter C. Dunphy, and Marshall S. Smith. 1966. The General Inquirer: A computer approach to content analysis. *MIT press.*

Hiroya Takamura and Manabu Okumura. 2009. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of EACL*, pages 781–789.

Yla R. Tausczik and James W. Pennebaker. 2007. The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology*, 29:24–54.

Simone Teufel and Marc Moens. 1997. Sentence extraction as a classification task. In *Proceedings of ACL*, pages 58–65.

Vishal Thapar, Ahmed A Mohamed, and Sanguthevar Rajasekaran. 2006. Consensus text summarizer based on meta-search algorithms. In *Proceedings of ISSPIT*, pages 403–407.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the NAACL-HLT*, pages 173–180.

Andranik Tumasjan, Timm Oliver Sprenger, Philipp G. Sandner, and Isabell M. Welpe. 2010. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In *Proceedings of ICWSM*, pages 178–185.

Gökhan Tür and Kemal Oflazer. 1998. Tagging english by path voting constraints. In *Proceedings of ACL-COLING: Short Papers*, pages 1277–1281.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394.

Peter Turney. 1997. Extraction of keyphrases from text: evaluation of four algorithms. *Technical Report ERB-1057.*

Peter Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336.

Hans Van Halteren, Jakub Zavrel, and Walter Daelemans. 1998. Improving data driven wordclass tagging by system combination. In *Proceedings of ACL-COLING*, pages 491–497.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618.

Xiaojun Wan and Jianwu Yang. 2008. Multi-document summarization using cluster-based link analysis. In *Proceedings of SIGIR*, pages 299–306.

Xiaojun Wan and Jianmin Zhang. 2014. CTSUM: extracting more certain summaries for news articles. In *Proceedings of SIGIR*, pages 787–796.

Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of ACL*, pages 552–559.

Xiaojun Wan, Ziqiang Cao, Furu Wei, Sujian Li, and Ming Zhou. 2015. Multi-document summarization via discriminative summary reranking. *arXiv preprint arXiv:1507.02062*.

Dingding Wang and Tao Li. 2010. Many are better than one: improving multi-document summarization via weighted consensus. In *Proceedings of SIGIR*, pages 809–810.

Dingding Wang and Tao Li. 2012. Weighted consensus multi-document summarization. *Information Processing & Management*, 48(3):513–523.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *Proceedings of ACL*, pages 1384–1394.

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

Janyce Wiebe and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. language resources and evaluation. In *Language Resources and Evaluation (formerly Computers and the Humanities)*, page 1(2).

Kristian Woodsend and Mirella Lapata. 2012. Multiple aspect summarization using Integer Linear Programming. In *Proceedings of EMNLP*, pages 233–243.

Han Xu, Eric Martin, and Ashesh Mahidadia. 2015. Extractive summarisation based on keyword profile and language model. In *Proceedings of NAACL*, pages 123–132.

Yinfei Yang and Ani Nenkova. 2014. Detecting information-dense texts in multiple news domains. In *Proceedings of AAAI*.

Wen-tau Yih, Joshua Goodman, and Vitor R. Carvalho. 2006. Finding advertising keywords on web pages. In *Proceedings of WWW*, pages 213–222.

Wen-tau Yih, Joshua Goodman, Lucy Vanderwende, and Hisami Suzuki. 2007. Multi-document summarization by maximizing informative content-words. In *Proceedings of IJCAI*, pages 1776–1782.

Hongyuan Zha. 2002. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *Proceedings of SIGIR*, pages 113–120.

Hui Zhang, Min Zhang, Chew Lim Tan, and Haizhou Li. 2009. K-best combination of syntactic parsers. In *Proceedings of EMNLP*, pages 1552–1560.

Kelly H. Zou, Kemal Tuncali, and Stuart G. Silverman. 2003. Correlation and simple linear regression. *Radiology*, 227(3):617–628.