ANSWERING COMPLEX QUESTIONS: SUPERVISED APPROACHES

SHEIKH SADID-AL-HASAN
Bachelor of Science in Computer Science and Information Technology
Islamic University of Technology (Bangladesh), 2005

A Thesis
Submitted to the School of Graduate Studies
of the University of Lethbridge
in Partial Fulfillment of the
Requirements for the Degree

MASTER OF SCIENCE

Department of Mathematics and Computer Science
University of Lethbridge
LETHBRIDGE, ALBERTA, CANADA

I dedicate this thesis to them without whom I could not be me as I am today.

Abstract

The term “Google” has become a verb for most of us. Search engines, however, have certain limitations. For example ask it for the impact of the current global financial crisis in different parts of the world, and you can expect to sift through thousands of results for the answer. This motivates the research in complex question answering where the purpose is to create summaries of large volumes of information as answers to complex questions, rather than simply offering a listing of sources. Unlike simple questions, complex questions cannot be answered easily as they often require inferencing and synthesizing information from multiple documents. Hence, this task is accomplished by the query-focused multi-document summarization systems. In this thesis we apply different supervised learning techniques to confront the complex question answering problem. To run our experiments, we consider the DUC-2007 main task.

A huge amount of labeled data is a prerequisite for supervised training. It is expensive and time consuming when humans perform the labeling task manually. Automatic labeling can be a good remedy to this problem. We employ five different automatic annotation techniques to build extracts from human abstracts using ROUGE, Basic Element (BE) overlap, syntactic similarity measure, semantic similarity measure and Extended String Subsequence Kernel (ESSK). The representative supervised methods we use are Support Vector Machines (SVM), Conditional Random Fields (CRF), Hidden Markov Models (HMM) and Maximum Entropy (MaxEnt). We annotate DUC-2006 data and use them to train our systems, whereas 25 topics of DUC-2007 data set are used as test data. The evaluation results reveal the impact of automatic labeling methods on the performance of the supervised approaches to complex question answering. We also experiment with two ensemble-based approaches that show promising results for this problem domain.

Acknowledgments

All praise be to the Almighty who gave me the opportunity to live in this world. Without his wish nothing could be made possible for me. Other than that, I believe that my ability to write this thesis is mainly rooted in a fortunate sequence of events, moral support from friends and family, and advice from scientists of extremely high caliber.

I have learned almost everything from my supervisor, Dr. Yllias Chali. He taught me how to start my journey in the field of Natural Language Processing. His continuous inspiration always acted as a fuel to work hard and to be focused in the right track. I am so lucky to get such a mentor as my supervisor. I am grateful to him.

I also thank my M.Sc. supervisory committee members Dr. Wendy Osborn and Dr. Sajjad Zahir for their valuable suggestions and guidance.

I received many helpful comments and suggestions from the anonymous reviewers during the submission process at different conferences and journals: ACL-IJCNLP-2009, CAI-2009, PACLING-2009, JAIR and IPM. I owe my sincere gratitude to them.

On the non-scientific side of my life, my parents never ceased to be my most devoted and faithful friends although they were more than a thousand miles away from me. They knew nothing about computer science, but what they taught me proved to be more essential and fundamental than all my University courses put together. I find no words to thank them for that.

I am also thankful to all my fellow office mates and friends: Shafiq, Kaisar, Tauhidul, Tarikul, Mahmud, Sangita, Chad and Chris for their encouragement and support. I will never forget Husam Ali for his unselfish help during all my difficult times.

Last but not the least, I thank NSERC and the University of Lethbridge for their financial assistance to carry out the whole work.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

As a result of the availability of all kind of information in the Web, we live in an easier world now. We do not bother to stay day after day in the library; rather a couple of hours of Web surfing does the job for us. The vast increase in both the amount of online data and the demand for access to different types of information have led researchers to a renewed interest in a broad range of Information Retrieval (IR) related areas such as question answering, topic detection and tracking, summarization, multimedia retrieval, chemical and biological informatics, text structuring and text mining. Automated Question Answering (QA) (Strzalkowski and Harabagiu, 2008) is the ability of a machine to answer questions, simple or complex, which are posed in ordinary human language. This is perhaps the most exciting technological development of the past six or seven years.

Traditional document retrieval systems cannot satisfy the end-users to have more direct access into relevant documents. So, Question Answering (QA) has received immense attention from the information retrieval, information extraction, machine learning, and natural language processing communities. The main goal of QA systems is to retrieve relevant answers to natural language questions from a collection of documents rather than employing keyword matching techniques to extract documents having keywords similar to a query. A well known QA system is the Korean Naver's Knowledge iN search[1], who were the pioneers in the community. This tool allows users to ask almost any question and get answers from other users. Naver's Knowledge iN now has roughly 10 times more entries than Wikipedia. It is used by millions of Korean web users on any given day. Some people say

[1] http://kin.naver.com/

Koreans are not addicted to the internet but to Naver (Chali, Joty, and Hasan, 2009). As of January 2008 the Knowledge Search database included more than 80 million pages of user-generated information. Another popular answer service is Yahoo! Answers[2] which is a community-driven knowledge market website launched by Yahoo!. It allows users to both submit questions to be answered and answer questions from other users. People vote on the best answer. The site gives members the chance to earn points as a way to encourage participation and is based on the Naver model. As of December 2009, Yahoo! Answers had 200 million users worldwide and more than 1 billion answers[3]. Google had a QA system[4] based on paid editors which was launched in April 2002 and fully closed in December 2006. The main limitation of these QA systems is that each relies on human expertise to help provide the answers. Our goal is to automate this process so that computers can do the same as those professional information analysts to give answers in response to the more complex questions that may include human language difficulties. This thesis is a small step towards such an ambitious goal.

QA research attempts to deal with a wide range of question types including: fact, list, definition, how, why, hypothetical, semantically-constrained, and cross-lingual questions. Some questions, which we call simple questions, are easier to answer. For example, the question: "Who is the president of Bangladesh?" asks for a person's name. This type of question (i.e. factoid) requires small snippets of text as the answer. Again, the question: "Which countries has Pope John Paul II visited?" is a sample of a list question asking only for a list of small snippets of text.

As a tool for finding documents on the web, search engines are proven to be adequate. Although there is no limitation in the expressiveness of the user in terms of query formu-

[2]http://answers.yahoo.com/

[3]http://yanswersblog.com/index.php/archives/2009/12/14/yahoo-answers-hits-200-million-visitors-worldwide/

[4]http://answers.google.com/

lation, certain limitations exist in what the search engine does with the query. Complex questions often seek multiple different types of information simultaneously and do not presuppose that one single answer can meet all of its information needs. For example, with a factoid question like: "What is the magnitude of the earthquake in Haiti?", it can be safely assumed that the submitter of the question is looking for a number. However, with complex questions like: "How is Haiti affected by the earthquake?", the wider focus of this question suggests that the submitter may not have a single or well-defined information need and therefore may be amenable to receiving additional supporting information that is relevant to some (as yet) undefined informational goal (Harabagiu, Lacatusu, and Hickl, 2006). Complex question answering tasks require multi-document summarization through an aggregated search, or a faceted search, that represents an information need which cannot be answered by a single document. For example, if we look for the comparison of the average number of years between marriage and first birth for women in the USA, Asia, and Europe, the answer is likely contained in multiple documents. Multi-document summarization is useful for this type of query and there is currently no tool on the market that is designed to meet this kind of information need.

Over the past few years, complex questions have been the focus of much attention in both the automatic Question Answering (QA) and Multi Document Summarization (MDS) communities. Typically, current complex QA evaluation systems including the 2004 AQUAINT Relationship QA Pilot[5], the 2005 Text Retrieval Conference (TREC) Relationship QA Task[6], and the TREC definition[7] return unstructured lists of candidate answers in response to a complex question. However, MDS evaluations (including the 2005, 2006 and 2007 Document Understanding Conference (DUC)[8]) have tasked systems with

[5]http://trec.nist.gov/data/qa/add_QAresources/README.relationship.txt
[6]http://trec.nist.gov/data/qa/2005_qadata/qa.05.guidelines.html
[7]http://trec.nist.gov/overview.html
[8]http://duc.nist.gov/

returning paragraph-length answers to complex questions that are responsive, relevant, and coherent.

In this thesis, we focus on a query-based extractive approach of summarization where a subset of the sentences from the original documents are chosen. The Document Understanding Conference (DUC) has been conducted by the National Institute of Standards and Technology (NIST) since 2001. Its goal is to further progress in automatic text summarization and enable researchers to participate in large-scale experiments in both the development and evaluation of summarization systems. DUC produces a series of summarization evaluations. The DUC-2007 main task was the same as the DUC 2006 task and modeled real-world complex question answering, in which a question cannot be answered by simply stating a name, date, quantity, etc. Given a topic and a set of 25 relevant documents, the task was to synthesize a fluent, well-organized 250-word summary of the documents that answers the question(s) in the topic statement. Successful performance on the task benefited from a combination of IR and NLP capabilities, including passage retrieval, compression, and generation of fluent text. In this thesis, we use the DUC-2007 main task to run our experiments.

The research in multi-document summarization domain that applies unsupervised learning techniques such as Expectation Maximization (EM) and K-means work on unlabeled data (Joty, 2008), where labeled data is a prerequisite for supervised systems. It is well known that supervised learning techniques often outperform unsupervised learning in terms of accuracy. That motivates us to use the supervised learning methodologies such as Support Vector Machines (SVM) (Cortes and Vapnik, 1995), Conditional Random Fields (CRF) (Lafferty, McCallum, and Pereira, 2001), Hidden Markov Models (HMM) (Conroy and O'Leary, 2001), Maximum Entropy (MaxEnt) (Ferrier, 2001) and ensemble methods (Dietterich, 2000) to combat the complex question answering problem.

1.2 Important Terms

Through out this thesis, we frequently refer to some important terms. We briefly introduce them here.

Abstract Summary: Human generated summary.

Extract Summary: Summary created by picking verbatim sentences from the original document that are mostly similar to the abstract summary.

Annotation/Labeling: The process of selecting important sentences from the original source document i.e. creation of extract summary.

Labeled Data: Original document sentences labeled with $+1$ meaning important enough for summary inclusion and -1, if opposite.

Training: To learn about important facts from the labeled data, accomplished by the supervised systems.

Testing: To predict labels for the previously unseen data set.

Balanced/Unbalanced Data: Balanced data contains an equal proportion of positive (summary) and negative (non-summary) sentences, whereas unbalanced data is comprised of an uneven proportion of positive and negative sentences.

Ensemble: Combined model of different classifiers intended to improve overall performance.

1.3 Overview of the Thesis

The block diagram in Figure 1.1 depicts the basic architecture of this thesis. The task of answering complex questions using a supervised methodology subsumes three major phases: Annotation/Labeling, Training and Testing. Given the original document sentences and their abstract summary, we produce the labeled data set through annotation. This phase in-

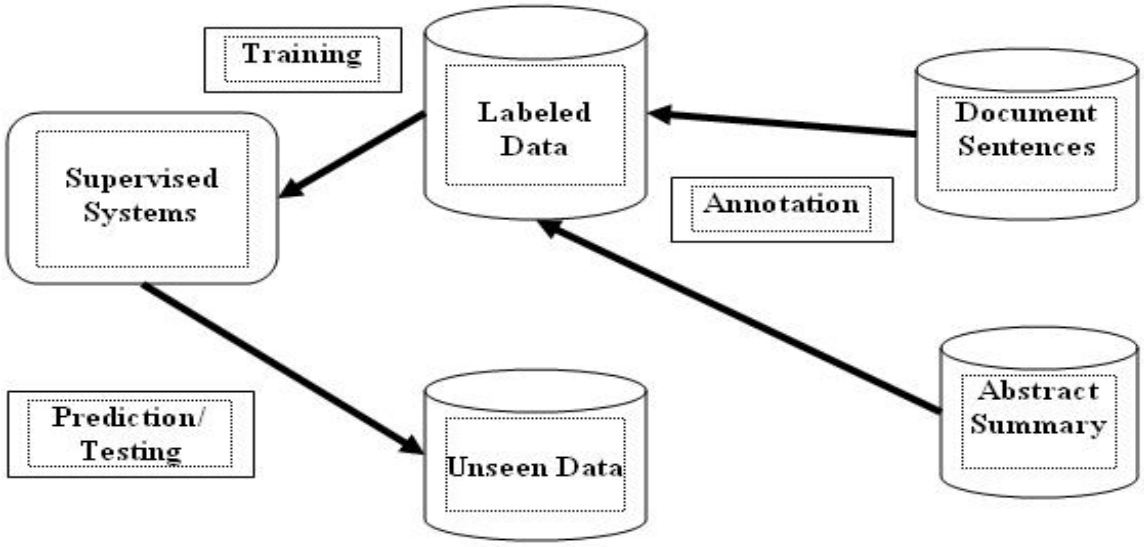

Figure 1.1: Supervised Approach

cludes the application of different textual similarity measurement techniques that compares the abstract summary sentences with the document sentences. The most similar sentences are labeled as positive (summary) and the less similar ones as negative (non-summary). In the second phase, supervised systems are trained with this large amount of labeled data. Finally, an unseen data set is presented before the learned model that predicts the labels in order to produce system generated extract summaries.

1.4 Research Questions

The better performance shown by supervised learning techniques over unsupervised approaches in terms of accuracy gives us a greater anticipation of applying these methods in the field of multi-document text summarization. The main goal of multi-document text summarization is to produce a single text as a compressed version of a set of documents with a minimum loss of relevant information. The research questions that we address in

this thesis are:

1. *Given a set of documents and their abstract summary, how to generate the extract summary? That is, from the document sentences how to find that a sentence is a good candidate to be extracted?*

2. *Given a set of labeled data that is a set of documents and their extract summary, how to learn the relationship between them and use this in order to predict the summary of an unseen data set?*

3. *Do automatic annotation techniques have any impact on supervised complex question answering?*

4. *How does the balanced or unbalanced labeled data affect the performance of supervised systems to complex question answering?*

5. *Do ensemble methods perform well in this problem domain?*

1.5 Contributions

This thesis contributes to the complex question answering task in the following ways:

Automatic Annotation We measure the similarity between the given abstract summary sentences and the original document sentences. Then we find the most similar sentences from the original document collection as the extract summary for a given topic. Thus we prepare five different types of labeled data to feed the supervised systems. We apply five different techniques to accomplish this: ROUGE similarity measure (Lin, 2004), Basic Element (BE) overlap (Hovy et al., 2006), syntactic similarity measure (Moschitti and Basili,

2006), semantic similarity measure (Moschitti et al., 2007), and Extended String Subsequence Kernel (ESSK) (Hirao et al., 2003).

Supervised Formulation We formulate the complex question answering problem in terms of supervised approaches. The representative supervised methods we use are Support Vector Machines (SVM), Conditional Random Fields (CRF), Hidden Markov Models (HMM), and Maximum Entropy (MaxEnt).

Impact of Automatic Annotation Techniques For the training of supervised systems, we use the five different types of labeled data. To show their impact, we extensively investigate the performance of the four classifiers to label unseen sentences as summary or non-summary sentence.

Balanced/Unbalanced Labeled Data During automatic annotation, we prepare a balanced labeled data set by treating 50% of sentences as extract summary sentences and the rest as non-summary sentences. The unbalanced labeled data set is made with only 30% of sentences as the summary sentences. We evaluate the performance of the supervised classifiers based on the use of balanced or unbalanced data during training.

Ensemble Methods We build two ensemble-based supervised systems and experiment their effectiveness to answer complex questions.

1.6 Thesis Outline

We give a chapter-by-chapter outline of the remainder of this thesis in this section.

Chapter 2 We give a detailed description of currently available summarization techniques. We then present our approach for the complex question answering task.

Chapter 3 We provide a general review of works performed previously in the automatic annotation area. Then we present an in-depth discussion of the five automatic annotation techniques used in this work.

Chapter 4 We take a closer look at the supervised approaches that were successfully used in different applications before and then we describe the theoretical aspects of them thoroughly.

Chapter 5 We discuss different summary evaluation techniques in this chapter.

Chapter 6 All the implementation related issues are discussed in detail.

Chapter 7 We present the experimental results and analyze them thoroughly.

Chapter 8 We conclude the thesis by identifying some future directions of our research.

1.7 Published Work

Some of the material presented in this thesis has been previously published. Chapter 3 to Chapter 7 expands on the materials published in (Chali, Hasan, and Joty, 2009b; Chali, Hasan, and Joty, 2009a; Chali, Hasan, and Joty, 2009c) and (Chali, Joty, and Hasan, 2009).

Chapter 2

Automatic Text Summarization

2.1 Introduction

In recent years, a great amount of attention has grown in both Question Answering (QA) and Multi-document Summarization (MDS) communities to deal with the query relevant summarization research (Carbonell et al., 2000). The synergy between text summarization and question answering systems worked as a catalyst behind this. Summarization is a process of condensing multiple source texts into one shorter version in response to complex questions, while Question Answering provides a means for focus in query-oriented summarization. Thus the boundaries between QA and MDS research communities are now beginning to blur. As complex questions cannot be answered using the same techniques that have successfully been applied to the answering of "factoid" questions, multi-document summarization techniques are applied to accomplish this task. Thus we focus more on the summarization aspects. The Information Retrieval phase for Question Answering falls outside the scope of this work. We assume the given set of documents as relevant for the given questions.

Text summarization is a good way to condense a large amount of information into a concise form by selecting the most important information and discarding redundant information. According to Mani (2001), automatic text summarization takes a partially-structured source text from multiple texts written about the same topic, extracts information content from it, and presents the most important content to the user in a manner sensitive to the user's needs. Although search engines do a remarkable job in searching through a heap of information, they have certain limitations. For example, if we ask for the impact of the current global financial crisis in different parts of the world, we can expect to sift through

thousands of results for the answer. The process of getting a desired answer to a complex question would speed up considerably when the summary of the given documents is also available. The technology of automatic summarization is critical in dealing with this kind of problems. In this chapter, we discuss different types and techniques of automatic text summarization and then we describe the approach we follow.

2.2 Types

The automatic text summarization task can be categorized into the following types:

1. Generic vs. Query-Oriented
2. Abstractive vs. Extractive
3. Single vs. Multi-Document

2.2.1 Generic vs. Query-Oriented

Generic summaries provide users with the overall sense of the document. A generic summary must contain the core information present in the document. Document understanding plays a key role here. Hence, most of the summaries created by the human beings are generic summaries. One of the most notable approaches to generic summarization has been introduced by Carbonell and Goldstein (1998) based on Maximal Marginal Relevance (MMR) that uses the vector-space model of text retrieval.

In recent years, attention has shifted from generic summarization toward query-based summarization. While a generic summary includes information which is central to the source documents, a query-oriented summary should formulate an answer to the query (Goldstein et al., 1999).

2.2.2 *Abstractive vs. Extractive*

An extract summary consists of sentences extracted from the document while an abstract summary may employ words and phrases that do not appear in the original document (Mani and Maybury, 1999). Abstract based summarization, as done by humans, involves reading and understanding an article, web site, document, etc. and then selecting the key points. Existing research has tried to emulate the human approach to the task with little success as several complicated factors such as word sense and grammatical structure have to be taken into consideration. The abstract summary that has all the characteristics of a good summary is the ultimate goal of automatic text summarization.

On the other hand, extract summarization is simpler than abstract summarization since the process involves assigning scores to the original sentences using some method and then picking the top-ranked sentences for the summary. Although this kind of summary may not be necessarily smooth or fluent, extractive summarization is currently a general practice among the automatic text summarization researchers for its simplicity.

2.2.3 *Single vs. Multi-Document*

The process of summarizing one document is termed as single document summarization whereas in multi-document summarization, multiple documents related to one main topic are used as sources. Single document summarization is useful in many situations such as summarizing e-mails, news articles or creating abstract of scientific research papers. Currently, multi-document summarization is of greater interest since the amount of information present in the web is becoming huge. For example, we can obtain the news about a single event from different sources in order to create the summary that provides multiple perspectives at the same time. Although the major challenges of multi-document summarization

such as completeness, readability, and conciseness are yet to be fulfilled, some web-based systems are already utilizing the potential of this technology. The Newsblaster[1] system automatically collects, clusters, categorizes, and summarizes news from several sites on the web, and helps users find the news of their greatest interest.

2.3 Techniques

The automatic text summarization area has gone through several changes with the development of techniques and requirements since the year 1950. Typically, we can divide the summarization approaches into the following categories:

1. Knowledge-based Methods
2. Classical Methods
3. Modern Methods

2.3.1 Knowledge-based Methods

It is always desirable to emulate the process of summarization as humans do it. To accomplish the summarization task automatically the machine needs to understand the source texts, pick out the important points and generate sentences from these points. The whole approach relies on both natural language understanding and generation. These methods are termed as knowledge-based methods (Ferrier, 2001). Although the model seems obvious, the major stages of it may subsume several substages. For instance in the language understanding phase, a process may exist for building individual sentence representations, followed by one for integrating these into a larger text representation, perhaps followed by

[1] http://newsblaster.cs.columbia.edu/

a further process for modifying the global text representation. This type of summarization is pretty hard for the machine to perform because they have to characterize a source text as a whole, capture its important content, where content is a matter both of information and its expression, and importance is a matter of what is essential as well as what is salient. Hence, other methods have also been investigated in order to produce automatic summaries.

2.3.2 Classical Methods

The methods that started the automatic text summarization research can be termed as the classical methods. Being motivated by the need to deal with the information overload problem, one of the first to perform such research was Luhn (1958). He realized the impracticality of trying to summarize more than one genre with one particular method and so, he chose to concentrate on texts in the scientific domain, where producing abstracts (written by humans) is a common practice. Luhn used simple statistical techniques to determine the most significant sentences of a document. These sentences were then extracted from the text and printed out together so that they became the summary, or more precisely the extract. Hence, the task became one of extraction and problematic issues fundamental to deep approaches such as natural language understanding and generation were reduced.

Methods to find features of the input text have been developed since Luhn's work. Edmundson (1969) weighted sentences based on four different methods: cue phrase, keyword (i.e., term frequency based), location, and title. He then evaluated each program by comparing against manually created extracts. He used a corpus based model, dividing the set of articles into a training and a test set. Edmundson found that the combination of cue-title-location features was the best, with location being the best individual feature and keywords the worst. These early ground-breaking systems acted as the pioneers to the modern summarization systems.

2.3.3 Modern Methods

The field of automatic text summarization has received immense attention from the researchers in the recent years. The vast increase of information in the web has acted as a fuel for this. Automatic summarization plays a central role in information retrieval. As the summarization tasks have changed, the methods to accomplish these have also kept pace. Below we discuss the most notable modern methods of automatic summarization.

Graph-based Recently, the graph-based methods, such as LexRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004), are applied successfully to generic, multi-document summarization. Erkan and Radev (2004) used the concept of graph-based centrality to rank a set of sentences for producing generic multi-document summaries. A similarity graph is produced for the sentences in the document collection. In the graph each node represents a sentence. An edge between two nodes measures the cosine similarity between the respective pair of sentences. The degree of a given node is an indication of how important the sentence is. A topic-sensitive LexRank is proposed in (Otterbacher, Erkan, and Radev, 2005). In this method, a sentence is mapped to a vector in which each element represents the occurrence frequency (TF*IDF) of a word. However, the major limitation of the TF*IDF approach is that it only retains the frequency of the words and does not take into account the sequence, syntactic and semantic structure, thus cannot distinguish between "The hero killed the villain" and "The villain killed the hero".

Latent Semantic Analysis (LSA) Latent Semantic Analysis (LSA) (Deerwester et al., 1990) is a fully automatic statistical technique to extract and infer relations of expected contextual usage of words in passages of discourse. The first step towards the application of LSA is to represent a document as a document-term matrix A, such that each row of matrix stands for a unique word present in the document and each column stands for a sentence.

Each entry A_{ij} represents the frequency of term i in document j. Gong and Liu (2001) have proposed a scheme for automatic text summarization using LSA. Their approach classifies the document into different topics and picks the dominant sentence from each dominant topic sequentially until the summary length is reached.

Lexical Chain A lexical chain is a sequence of related words in the text, spanning short (adjacent words or sentences) or long distances (entire text). A chain is independent of the grammatical structure of the text and in effect it is a list of words that captures a portion of the cohesive structure of the text. Computing the lexical chains in a document is one technique that can be used to identify the central theme of a document. This in turn leads to the identification of the key section(s) of the document which can then be used for summarization purposes. The summarization systems based on lexical chain first extract the nouns, compound nouns and named entities as candidate words (Li et al., 2007), (Kolla, 2004). The systems rank sentences using a formula that involves a) the lexical chain, b) keywords from query and c) named entities. For example, (Li et al., 2007) uses the following formula:

$$Score = \alpha P(chain) + \beta P(query) + \gamma P(nameentity)$$

where $P(chain)$ is the sum of the scores of the chains whose words come from the candidate sentence, $P(query)$ is the sum of the co-occurrences of key words in a topic and the sentence, and $P(nameentity)$ is the number of name entities existing in both the topic and the sentence. The three coefficients α, β and γ are set empirically. Then the top ranked sentences are selected to form the summary.

QA System Typically, in a summarization system that is based on a question answering system (Molla and Wan, 2006), the topic sentences are converted to a sequence of questions

as the underlined QA system is designed to answer only simple (i.e. factoid, list) questions. The QA system normalizes and classifies the questions, and finds the candidate answers along with the sentences in which the answers appeared. Instead of extracting the exact terms as answers, the systems extract the sentences for each of the questions in the topic to form the summary.

Machine Learning Techniques Machine Learning (ML) is concerned with the design and development of algorithms and techniques that allow computers to learn. The major focus of machine learning research is to extract information from data automatically, by computational and statistical methods. The major challenge in summarization lies in distinguishing the more informative parts of a document from the less informative ones. In the 1990s, with the advent of machine learning techniques in NLP, a series of seminal publications appeared that employed statistical techniques to produce document extracts for single document summarization.

While initially most systems assumed feature independence and relied on naive-Bayes methods, others have focused on the choice of appropriate features and on learning algorithms that make no independence assumptions. Other significant approaches involved hidden Markov models and log-linear models to improve extractive summarization (Das and Martins, 2007). Single document summarization systems using Support Vector Machines (SVMs) demonstrated good performance for both Japanese (Hirao et al., 2002a) and English documents (Hirao et al., 2002b). Hirao et al. (2003) showed the effectiveness of their multiple document summarization system employing SVMs for sentence extraction. Conroy and O'Leary (2001) used two kinds of states, where one kind corresponds to the summary states and the other corresponds to non-summary states. The motivation of applying CRF in text summarization came from observations on how humans summarize a document by posing the problem as a sequence labeling problem (Shen et al., 2007). The

statistical technique such as Maximum Entropy (MaxEnt) works in a way that assumes nothing about the information of which it has no prior knowledge (Ferrier, 2001). Joty (2008) experimented with both empirical and unsupervised machine learning approaches (K-means and Expectation Maximization (EM) algorithms) to summarize texts.

2.4 Our Approach

In this thesis, we focus our research on query-oriented, extractive, multi-document summarization in order to combat the complex question answering problem such as the one defined in the DUC-2007 main task. We apply supervised machine learning techniques: Support Vector Machines (SVM), Hidden Markov Models (HMM), Conditional Random Fields (CRF), and Maximum Entropy (MaxEnt) to perform the task of automatic summarization. As supervised systems rely on learning from a vast amount of labeled data, we automatically annotate DUC-2006 data using five text similarity measurement techniques: ROUGE similarity measure (Lin, 2004), Basic Element (BE) overlap (Hovy et al., 2006), syntactic similarity measure (Moschitti and Basili, 2006), semantic similarity measure (Moschitti et al., 2007), and Extended String Subsequence Kernel (ESSK) (Hirao et al., 2003). We also experiment with supervised ensemble-based approaches that combine the individual decisions of the classifiers. In supervised learning, the classifier is typically trained on data pairs defined by feature vectors and corresponding class labels. Besides using different query-related features, we incorporate some important features from the classical methods of summarization.

2.5 Summary

We discussed different types and approaches to automatic text summarization in this chapter. Next chapter will present details on the automatic annotation techniques that we used to generate huge amount of labeled data required for supervised training.

Chapter 3

Automatic Annotation Techniques

3.1 Introduction

Annotated corpora are essential for most branches of computational linguistics, including automatic text summarization. Within computational linguistics, annotated corpora are normally considered as a gold standard, and are used to train machine learning algorithms and evaluate the performance of automatic summarization methods (Orasan, 2005). So, for supervised learning techniques, a huge amount of annotated or labeled data is required as a precondition. The decision as to whether a sentence is important enough to be annotated can be made either by humans or by programs. When humans are employed in the process, producing such a large labeled corpora becomes time consuming and expensive. There comes the necessity of using automatic methods to align sentences with the intention to build extracts from abstracts.

Annotation has been employed in automatic summarization since the late 1960s when Edmundson used one in the evaluation process. In order to produce the annotated corpus, Edmundson asked humans to identify the important sentences in each text from a collection of 200 scientific documents (Edmundson, 1969). Given that identification of important sentences is very subjective and difficult, Kupiec, Pedersen, and Chen (1995) took advantage of human produced abstracts, and asked annotators to align sentences from the document with sentences from there. In the automatic annotation area, Banko et al. (1999) proposed a method based on sentence similarity using a bag-of-words (BOW) representation. For each sentence in the given abstract, the corresponding source sentence is determined by combining the similarity score and heuristic rules. Marcu (1999) treated a sentence as a set of units that correspond to clauses and defines similarity between units based on BOW rep-

resentation. Jing and McKeown (1999) proposed a bigram-based similarity approach using the Hidden Markov Model. Barzilay (2003) combines edit distance and context information around sentences for annotation. However, as these methods are strongly influenced by word order, disagreement between source and abstract summary sentences leads to failure. Toutanova et al. (2007) used the ROUGE[1] (Lin, 2004) toolkit to produce labeled data automatically. The "head-modifier-relation" triples, typically considered as Basic Elements (BE), can help deciding whether any two units match or not considerably more easily than with longer units (Hovy, Lin, and Zhou, 2005).

Approaches in Recognizing Textual Entailment, Sentence Alignment and Question Answering use syntactic and/or semantic information in order to measure the similarity between two textual units. Corresponding sentences can be parsed into syntactic trees using a syntactic parser. The similarity between the two trees can be calculated using the tree kernel (Collins and Duffy, 2001). Shallow semantic representations could prevent the sparseness of deep structural approaches and overcome the weakness of BOW models (Moschitti et al., 2007).

Hirao et al. (2004) represented the sentences using Dependency Tree Path (DTP) to incorporate syntactic information. They applied String Subsequence Kernel (SSK) to measure the similarity between the DTPs of two sentences and introduced Extended String Subsequence Kernel (ESSK) considering all possible senses to each word for building extracts from abstracts. Their method was effective. However, the fact that they did not disambiguate word senses cannot be disregarded.

The textual similarity measurement techniques of ROUGE similarity measure, Basic Element (BE) overlap, syntactic similarity measure, semantic similarity measure and Extended String Subsequence Kernel (ESSK) are reimplemented in this research to do sen-

[1]It is widely used for automatic summarization evaluation to measure the summary quality by counting overlapping units between the candidate summary and the reference summary.

tence annotation and explained in more detail in the later sections.

3.2 ROUGE Similarity Measures

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is an automatic tool to determine the quality of a summary by comparing it to reference summaries using a collection of measures (Lin, 2004). The measures count the number of overlapping units such as n-gram, word-sequences, and word-pairs between the extract and the abstract summaries. The ROUGE measures considered are: ROUGE-N (N=1,2,3,4), ROUGE-L, ROUGE-W and ROUGE-S.

ROUGE-N is n-gram recall between a candidate summary and a set of reference summaries which is computed as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in ReferenceSummaries} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in ReferenceSummaries} \sum_{gram_n \in S} Count(gram_n)}$$

where, n is the length of $n-grams$ and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. In case of multiple abstracts, pairwise summary-level ROUGE-N between a candidate summary s and every reference r_i, in the reference set is computed. Final ROUGE-N score is then obtained by taking the maximum of the summary-level ROUGE-N scores:

$$\text{ROUGE-N}_{multi} = argmax_i(\text{ROUGE-N}(r_i, s))$$

The ROUGE-L (Longest Common Subsequence-LCS) score between two summary sentences r of length m and s of length n (assuming r is a reference summary sentence and

s is a peer summary sentence) can be computed as follows (Lin, 2004):

$$\begin{aligned} R_{lcs} &= \frac{LCS(r,s)}{m} \\ P_{lcs} &= \frac{LCS(r,s)}{n} \\ F_{lcs} &= \frac{P_{lcs}R_{lcs}}{\alpha R_{lcs} + (1-\alpha)P_{lcs}} \end{aligned}$$

where P is the precision, R is recall and F-measure combines precision and recall into a single measure. ROUGE-W (Weighted Longest Common Subsequence-WLCS) provides an improvement to the basic LCS method of computation by using the function $f(n)$ to credit the sentences having the consecutive matches of words. WLCS can be calculated as follows:

$$\begin{aligned} R_{wlcs} &= f^{-1}\left(\frac{WLCS(X,Y)}{f(m)}\right) \\ P_{wlcs} &= f^{-1}\left(\frac{WLCS(X,Y)}{f(n)}\right) \\ F_{wlcs} &= \frac{P_{wlcs}R_{wlcs}}{\alpha R_{wlcs} + (1-\alpha)P_{wlcs}} \end{aligned}$$

The ROUGE-S (Skip bi-gram) score between the candidate summary sentence S of length m and the reference summary sentence R of length n can be computed as follows:

$$
\begin{aligned}
R_{skip_2} &= \frac{SKIP_2(S,R)}{C(m,2)} \\
P_{skip_2} &= \frac{SKIP_2(S,R)}{C(n,2)} \\
F_{lcs} &= \frac{P_{skip_2} R_{skip_2}}{\alpha R_{skip_2} + (1-\alpha) P_{skip_2}}
\end{aligned}
$$

where, $SKIP_2(S,Q)$ is the number of skip bi-gram (any pair of words in their sentence order, allowing for arbitrary gaps) matches between S and R, and α is a constant that determines the importance of precision and recall. C is the combination function. ROUGE-S is extended with the addition of unigram as counting unit which is called ROUGE-SU (Lin, 2004).

We assume each individual document sentence as the extract summary and calculated its ROUGE similarity scores with the corresponding abstract summaries. Thus an average ROUGE score is assigned to each sentence in the document. We choose the top N sentences based on ROUGE scores to have the label $+1$ (summary sentences) and the rest to have the label -1 (non-summary sentences).

3.3 Basic Element (BE) Overlap Measure

According to Hovy et al. (2006), Basic Elements (BEs) are defined as:

- the head of a major syntactic constituent (noun, verb, adjective or adverbial phrases), expressed as a single item, or
- a relation between a head-BE and a single dependent, expressed as a triple: (head|modifier|relation).

With BE represented as a head-modifier-relation triple, one can quite easily decide whether any two units match or not considerably more easily than with longer units.

We use the syntactic parser Minipar[2] to produce a parse tree. Then a set of "cutting rules" are employed to extract only the valid BEs from the tree. We extract BEs for the sentences in the document collection using BE package 1.0 distributed by ISI[3]. Once we obtain the BEs for a sentence, we compute the Likelihood Ratio (LR) for each BE (Hovy, Lin, and Zhou, 2005). The LR score of each BE is an information theoretic measure that represents the relative importance in the BE list from the document set that contains all the sentences to be aligned. Sorting the BEs according to their LR scores produces a BE-ranked list. Our goal is to find similarity between document sentences and reference summary sentences. The ranked list of BEs in this way contains important BEs at the top which may or may not be relevant to the abstract summary sentences. We filter those BEs by checking whether they contain any word that matches an *abstract sentence word* or a *related word* (i.e. synonyms, hypernyms, hyponyms and gloss words which are found using Wordnet (Fellbaum, 1998)). For each abstract sentence, we assign a score to every document sentence as the sum of its filtered BE scores divided by the number of BEs in the sentence. Thus, every abstract sentence contributes to the BE score of each document sentence and we select the top N number of sentences based on average BE scores to have the label $+1$ (summary) and rest to have the label -1 (non-summary).

3.4 Syntactic Similarity Measure

Word dependencies having an important role in finding similarity between two texts can be discovered using a syntactic parser. Syntactic parsing is analyzing a sentence using the

[2]Available at http://www.cs.ualberta.ca/ lindek/minipar.htm
[3]BE website:http://www.isi.edu/ cyl/BE

grammar rules. One method to tag word dependencies is by using the Charniak parser[4]. Pasca and Harabagiu (2001) demonstrated that with the syntactic form one can see which words depend on other words. There should be a similarity between the words that are dependent in the reference summary sentence and the dependency between words of the document sentence. Syntactic feature is used successfully in *question answering* earlier (Zhang and Lee, 2003; Moschitti et al., 2007; Moschitti and Basili, 2006).

In order to calculate the syntactic similarity between the abstract sentence and the document sentence, we first parse the corresponding sentences into syntactic trees using a parser like Charniak (1999). Then we calculate the similarity between the two trees using the *tree kernel* (Collins and Duffy, 2001). We convert each parenthetic representation generated by the Charniak parser into its corresponding tree and give the trees as input to the tree kernel functions for measuring the syntactic similarity. The tree kernel of two syntactic trees T_1 and T_2 is actually the inner product of $v(T_1)$ and $v(T_2)$:

$$TK(T_1, T_2) = v(T_1).v(T_2) \tag{3.1}$$

We define the indicator function $I_i(n)$ to be 1 if the sub-tree i is seen rooted at node n and 0 otherwise. It follows:

$$v_i(T_1) = \sum_{n_1 \in N_1} I_i(n_1)$$
$$v_i(T_2) = \sum_{n_2 \in N_2} I_i(n_2)$$

[4]available at ftp://ftp.cs.brown.edu/pub/nlparser/

where, N_1 and N_2 are the set of nodes in T_1 and T_2 respectively. So, we can derive:

$$\begin{aligned} TK(T_1,T_2) &= v(T_1).v(T_2) \\ &= \sum_i v_i(T_1)v_i(T_2) \\ &= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \sum_i I_i(n_1)I_i(n_2) \\ &= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} C(n_1,n_2) \end{aligned} \tag{3.2}$$

where, we define $C(n_1,n_2) = \sum_i I_i(n_1)I_i(n_2)$. The TK (tree kernel) function gives the similarity score between the abstract sentence and the document sentence based on the syntactic structure. Each abstract sentence contributes a score to the document sentences and the top N number of sentences are selected to be annotated as $+1$ and the rest as -1 based on the average of similarity scores.

3.5 Semantic Similarity Measure

Shallow semantic representations, bearing a more compact information, can prevent the sparseness of deep structural approaches and the weakness of BOW models (Moschitti et al., 2007). Initiatives such as PropBank (PB) (Kingsbury and Palmer, 2002) made it possible to design accurate automatic Semantic Role Labeling (SRL) systems (Hacioglu et al., 2003). So, attempting an application of SRL to automatic annotation seems natural, as similarity of an abstract sentence with a document sentence relies on a deep understanding of the semantics of both. For example, let us consider the PB annotation:

```
[ARG0 all] [TARGET use]
[ARG1 the french franc]
```

```
[ARG2 as their currency]
```

Such annotation can be used to design a shallow semantic representation that can be matched against other semantically similar sentences, e.g.

```
[ARG0 the Vatican] [TARGET uses]
[ARG1 the Italian lira]
[ARG2 as their currency]
```

To experiment with semantic structures, we parse the corresponding sentences semantically using a Semantic Role Labeling (SRL) system like ASSERT[5]. ASSERT is an automatic statistical semantic role tagger, that can annotate naturally occurring text with semantic arguments. When presented with a sentence, it performs a full syntactic analysis of the sentence, automatically identifies all the verb predicates in that sentence, extracts features for all constituents in the parse tree relative to the predicate, and identifies and tags the constituents with the appropriate semantic arguments. We represent the annotated sentences using tree structures called semantic trees (ST). In the semantic tree, arguments are replaced with the most important word, often referred to as the semantic head. We look for noun, then verb, then adjective, then adverb to find the semantic head in the argument. As in tree kernels (Section 3.4), common substructures cannot be composed by a node with only some of its children as an effective ST representation would require, Moschitti et al. (2007) solved this problem by designing the Shallow Semantic Tree Kernel (SSTK) which allows to match portions of a ST. The SSTK changes the ST by adding *SLOT* nodes, which provides a fixed number of slots, possibly filled with *null* arguments that encode all possible predicate arguments. The slot nodes are used in such a way that the adopted TK function can generate fragments containing one or more children. These changes generate a new C which, when substituted (in place of the original C) in Eq. 3.2, gives the new

[5] available at http://cemantix.org/assert

SSTK function that yields the similarity score between an abstract sentence and a document sentence based on semantic structure. Thus, each document sentence gets a semantic similarity score corresponding to each abstract sentence and then the top N number of sentences are selected to be labeled as $+1$ and the rest as -1 on the basis of average similarity scores.

3.6 Extended String Subsequence Kernel (ESSK)

The ESSK, a similarity measure is a simple extension of the Word Sequence Kernel (WSK) (Cancedda et al., 2003) and String Subsequence Kernel (SSK) (Lodhi et al., 2002). WSK receives two sequences of words as input and maps each of them into a high-dimensional vector space. WSK's value is just the inner product of the two vectors. But, WSK disregards synonyms, hyponyms, and hypernyms. On the otherhand, SSK measures the similarity between two sequences of "alphabets". In ESSK, each "alphabet" in SSK is replaced by a disjunction of an "alphabet" and its alternative (Hirao et al., 2003). Here, each word in a sentence is considered an "alphabet", and the alternative is its disambiguated sense that we find using the WSD (Word Sense Disambiguation) System of Chali and Joty (2007). The use of word sense yields flexible matching even when paraphrasing is used for summary sentences (Hirao et al., 2004).

We calculate the similarity score $\mathrm{Sim}(T_i, U_j)$ using ESSK where T_i denotes abstract sentence and U_j stands for document sentence. Formally, ESSK is defined as follows (Hirao et al., 2004):

$$K_{esk}(T,U) = \sum_{m=1}^{d} \sum_{t_i \in T} \sum_{u_j \in U} K_m(t_i, u_j)$$

$$K_m(t_i,u_j) = \begin{cases} val(t_i,u_j) & \text{if } m = 1 \\ K_{m-1}^{'}(t_i,u_j) \cdot val(t_i,u_j) & \end{cases}$$

Here, $K_m^{'}(t_i,u_j)$ is defined below. t_i and u_j are nodes of T and U, respectively. Each node includes a word and its disambiguated sense[6] (Chali and Joty, 2007). The function $val(t,u)$ returns the number of attributes common to the given nodes t and u.

$$K_m^{'}(t_i,u_j) = \begin{cases} 0 & \text{if } j = 1 \\ \lambda K_m^{'}(t_i,u_{j-1}) + K_m^{''}(t_i,u_{j-1}) & \end{cases}$$

Here λ is the decay parameter for the number of skipped words. $K_m^{''}(t_i,u_j)$ is defined as:

$$K_m^{''}(t_i,u_j) = \begin{cases} 0 & \text{if } i = 1 \\ \lambda K_m^{''}(t_{i-1},u_j) + K_m(t_{i-1},u_j) & \end{cases}$$

Finally, the similarity measure is defined after normalization as below:

$$sim_{esk}(T,U) = \frac{K_{esk}(T,U)}{\sqrt{K_{esk}(T,T)K_{esk}(U,U)}}$$

Indeed, this is the similarity score we assigned to each document sentence for each abstract sentence and in the end, the top N number of sentences are selected to be annotated

[6]We use a dictionary based disambiguation approach assuming one sense per discourse. We use WordNet (Fellbaum, 1998) to accomplish this.

as $+1$ and the rest as -1 based on average similarity scores.

3.7 Summary

In this chapter, we discussed the necessity of automatic annotation to produce large amount of labeled data. Then, we described five automatic annotation techniques that are used in this research. Next chapter will focus on the supervised approaches that we used to solve the complex question answering problem.

Chapter 4

Supervised Learning Approaches

4.1 Introduction

Supervised classifiers are typically trained on data pairs, defined by feature vectors and corresponding class labels. On the other hand, unsupervised approaches rely on heuristic rules that are pretty difficult to generalize (Shen et al., 2007). Supervised extractive summarization can often be regarded as a two-class classification problem that treats summary sentences as positive samples and non-summary sentences as negative samples. Given the features of a sentence, a machine-learning based classification model can judge how likely the sentence is important to be in the summary (Wong, Wu, and Li, 2008).

The Hidden Markov Model (HMM) requires a careful feature selection to achieve high accuracy (Kudo and Matsumoto, 2001) while bearing fewer assumptions of independence (Conroy and O'Leary, 2001). HMMs had been successfully applied to many data labeling tasks such as POS tagging (Kupiec, 1992), shallow parsing (Pla, Molina, and Prieto, 2000) and speech recognition (Rabiner and Juang, 1993). Conroy and O'Leary (2001) used the HMM method denoting two kinds of states, where one kind corresponds to the summary states and the other corresponds to the non-summary states. Given a new cluster of documents, they calculated the probability of a sentence to be in a summary state. Finally, the trained model was used to select the most likely summary sentences.

The statistical technique Maximum Entropy (MaxEnt) works in a way assuming nothing about the information of which it has no prior knowledge (Ferrier, 2001). Models based on maximum entropy are well suited to the sentence extraction task along with providing competitive results on a variety of language tasks (Berger, Pietra, and Pietra, 1996). Ferrier (2001) applied the MaxEnt technique to text summarization and found that the maximum

entropy classifier produces better results than the naive Bayes technique.

Conditional Random Fields (CRF) tend to carry out the summarization task in a discriminative manner (Shen et al., 2007). The motivation of applying CRF in text summarization came from observations on how humans summarize a document by posing the problem as a sequence labeling problem. Shen et al. (2007) showed the effectiveness of CRF by applying it to a generic single-document extraction task.

On the other hand, Support Vector Machines (SVM) take a strategy that maximizes the margin between critical samples and the separating hyperplane for efficient classification (Vapnik, 1998). By introducing the Kernel function, SVMs handle non-linear feature spaces, and carry out the training considering combinations of more than one feature. In the field of natural language processing, SVMs are applied to text categorization and syntactic dependency structure analysis, and are reported to have achieved higher accuracy than previous approaches (Joachims, 1998). Single document summarization systems using SVMs demonstrated good performance for both Japanese (Hirao et al., 2002a) and English documents (Hirao et al., 2002b). Hirao et al. (2003) showed effectiveness of their multiple document summarization system employing SVMs for sentence extraction.

At present, one of the most active research areas in supervised learning is the methods for constructing good ensemble of classifiers which needs the sub-classifiers to differentiate greatly (Qi and Huang, 2007). Ensemble techniques are in the focus of the researchers over the years as different methods for constructing good ensembles are developed (Dietterich, 2000). There have been many studies on the idea of creating multiple models on the training data and combining the predictions of each model. Several ensemble approaches have been successfully applied to text classification tasks, such as boosting, Error-Correcting Output Codes (ECOC), hierarchical mixture model and automated survey coding (Freund and Schapire, 1995; Dietterich and Bakiri, 1995; Toutanova et al., 2001). Alternative approaches such as stacking and earlier metaclassifier approaches (Bennett, Dumais, and

Horvitz, 2002) do not partition the data, but rather combine classifiers each of which attempts to classify all data over the entire category space. With the same learning algorithm, different classifiers can be generated by manipulating the training set, manipulating the input features, manipulating the output targets or injecting randomness in the learning algorithm (Dietterich, 2000). Ensemble approaches such as a stacking method was proposed by Wolpert (1992) and Meta Decision Trees was proposed by Todorovski and Dzeroski (2000). Yan et al. (2003) constructed SVM ensembles for rare class predictions in scene classification building individual training sets by combining a subset of negative data with all the positive data, and aggregate the output value of each classifier. Hoi and Lyu (2004) provided an algorithm called group-based relevance feedback with SVM ensemble successfully. A SVM ensemble based on majority voting mechanism was proposed in 2004 to do a classification experiment on the Hepatitis and Ionosphere data set of the UCI benchmark database that found the error rate lowered 10% averagely compared to a single classifier (Wei and Zhang, 2004). Nguyen et al. (2005) used a boosting based support vector ensemble to achieve good performance in summarizing text from a Vietnamese corpus. Rare class text categorization was successfully performed with SVM ensemble by Silva and Ribeiro (2006) where the learning strategy uses the separating margin as differentiating factor on positive classifications. SVM ensembles were also effectively applied in remote sensing classification (Qi and Huang, 2007).

We employ these supervised approaches to combat the complex question answering problem. Next sections give a detailed description of these approaches.

4.2 Hidden Markov Models (HMM)

HMMs are a form of generative model, that assign a joint probability $p(x,y)$ to pairs of observation and label sequences, x and y respectively (Wallach, 2002). Each observation

sequence (here, sentence sequence)is considered to have been generated by a sequence of state transitions, beginning in some start state and ending when some predesignated final state is reached. At each state an element of the observation sequence is stochastically generated, before moving to the next state. For any observation sequence, the sequence of states that best accounts for that observation sequence is essentially hidden from an observer and can only be viewed through the set of stochastic processes that generate an observation sequence. The principle of identifying the most state sequence that best accounts for an observation sequence forms the foundation underlying the use of finite-state models for labeling sequential data.

Formally, an HMM is fully defined by

- A finite set of states S.
- A finite output alphabet X.
- A conditional distribution $P\left(s^{'}|s\right)$ representing the probability of moving from state s to state $s^{'}$, where $s, s^{'} \in S$
- An observation probability distribution $P(x|s)$ representing the probability of emitting observation x when in state s, where $x \in X$ and $s \in S$.
- An initial state distribution $P(s), s \in S$.

A HMM may be represented as a directed graph G with nodes S_t and X_t representing the state of the HMM (or label) at time t and the observation at time t, respectively. This structure is shown in Figure 4.1 (Wallach, 2002).

This representation of a HMM clearly highlights the conditional independence relations within a HMM. Specifically, the probability of the state at time t depends only on the state at time $t-1$. Similarly, the observation generated at time t only depends on the state of the model at time t.

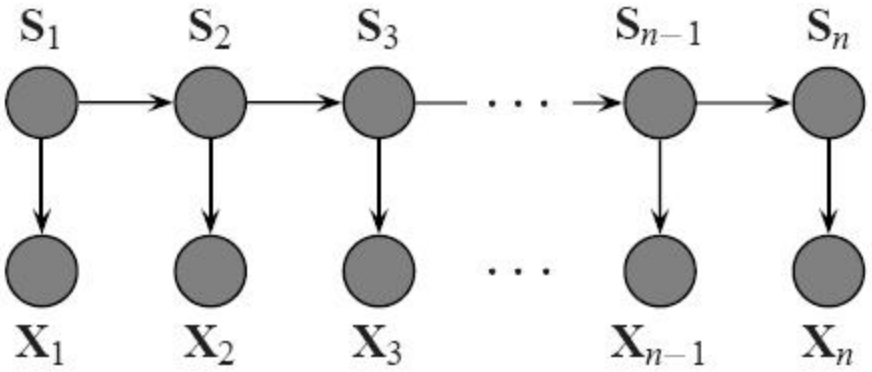

Figure 4.1: HMM structure

The conditional independence relations, combined with the probability chain rule, may be used to factorize the joint distribution over a state sequence s and observation sequence x into the product of a set of conditional probabilities:

$$p(s,x) = p(s_1)\,p(x_1|s_1)\prod_{t=2}^{n} p(s_t|s_{t-1})\,p(x_t|s_t) \tag{4.1}$$

Finding the optimal state sequence given the observation sequence and the model is most efficiently performed using a dynamic programming technique known as Viterbi alignment (Rabiner, 1989).

4.3 Maximum Entropy (MaxEnt)

The maximum entropy approach is a novel method for the task of sentence extraction. The main principle of the MaxEnt method is to model all that is known and assume nothing about that which is unknown. In other words, given a collection of facts, the model must be

consistent with all the facts, but otherwise act as uniformly as possible (Berger, Pietra, and Pietra, 1996). One advantage of this form of statistical inference is that we only constrain the model of our data by the information that we do know about the task, i.e. we do not assume anything about information of which we have no knowledge. Another advantage is that the information we use to constrain the model is in no way restricted so we can encode whatever linguistic information we want via the features. However, a disadvantage of this approach is that, although the maximum entropy approach may make good predictions, we cannot interpret the individual elements that cause the behavior of the system as a large number of features tend to be used in the approach and hence the output cannot be used to interpret all of these separately (Ferrier, 2001).

MaxEnt models can be termed as multinomial logistic regression if they are to classify the observations into more than two classes (Jurafsky and Martin, 2008). However, in this research, we used the MaxEnt model to classify the sentences into two classes: summary or non-summary. The parametric form for the maximum entropy model is as follows (Nigam, Lafferty, and McCallum, 1999):

$$P(c|s) = \frac{1}{Z(s)} exp\left(\sum_i \lambda_i f_i\right) \tag{4.2}$$

$$Z(s) = \sum_c exp\left(\sum_i \lambda_i f_i\right) \tag{4.3}$$

Here, c is the class label and s is the item we are interested in labeling that is the sentences here. Z is the normalization factor that is just used to make the exponential into a true probability. Each f_i is a feature with the associated weight λ_i which can be determined by numerical optimization techniques in absence of a closed form solution.

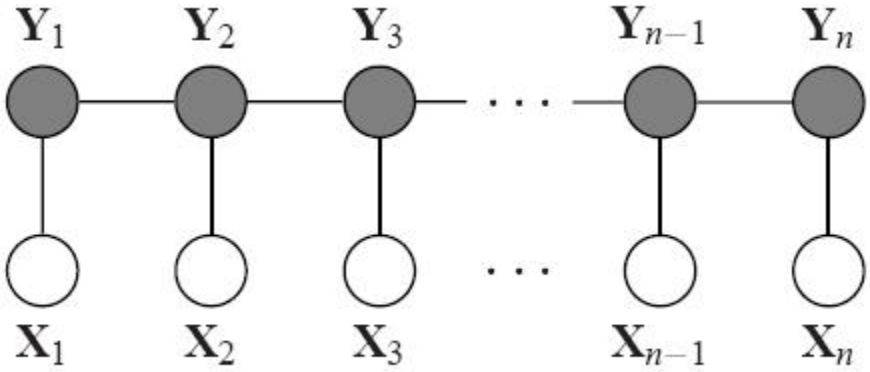

Figure 4.2: CRF model structure

4.4 Conditional Random Fields (CRF)

To reap the benefits of using a conditional probabilistic framework for labeling sequential data and simultaneously overcome the label bias problem, Lafferty, McCallum, and Pereira (2001) introduced CRFs. So, CRFs are conditional probabilistic sequence models, however, rather than being directed graphical models, CRFs are undirected graphical models (Wallach, 2002). This allows the specification of a single joint probability distribution over the entire label sequence given the observation sequence, rather than defining per-state distributions over the next states given the current state. The conditional nature of the distribution over label sequences allows CRFs to model real-world data in which the conditional probability of a label sequence can depend on non-independent, interacting features of the observation sequence. In addition to this, the exponential nature of the distribution chosen by Lafferty, McCallum, and Pereira (2001) enables features of different states to be traded off against each other, weighting some states in a sequence as being more important than others.

Figure 4.2 shows the linear chain model structure of CRF (Wallach, 2002).

CRF allows the specification of a single joint probability distribution over the entire

label sequence given the observation sequence, rather than defining per-state distributions over the next states given the current state. Given an observation sequence (sentence sequence here) $X = (x_1, \cdots, x_T)$ and the corresponding state sequence $Y = (y_1, \cdots, y_T)$, the probability of Y conditioned on X defined in CRFs, $P(Y|X)$, is as follows:

$$\frac{1}{Z_X} exp\left(\sum_{i,k} \lambda_k f_k(y_{i-1}, y_i, X) + \sum_{i,l} \mu_l g_l(y_i, X)\right) \tag{4.4}$$

where Z_X is the normalization constant that makes the probability of all state sequences sum to one; $f_k(y_{i-1}, y_i, X)$ is an arbitrary feature function over the entire observation sequence and the states at positions i and $i-1$ while $g_l(y_i, X)$ is a feature function of state at position i and the observation sequence; λ_k and μ_l are the weights learned for the feature functions f_k and g_l, reflecting the confidence of the feature functions (Shen et al., 2007).

4.5 Support Vector Machines (SVM)

SVM is a powerful methodology for solving machine learning problems introduced by Vapnik (Cortes and Vapnik, 1995) based on the Structural Risk Minimization principle. In the classification problem, the SVM classifier typically follows from the solution to a quadratic problem. SVM finds the separating hyperplane that has maximum margin between the two classes in case of binary classification. Separating the examples with a maximum margin hyperplane is motivated by the results from statistical learning theory, which states that a learning algorithm, to achieve good generalization, should minimize both the empirical error and also the "capacity" of the functions that the learning algorithm implements.

Figure 4.3 shows the conceptual structure of SVM. Training samples each of which

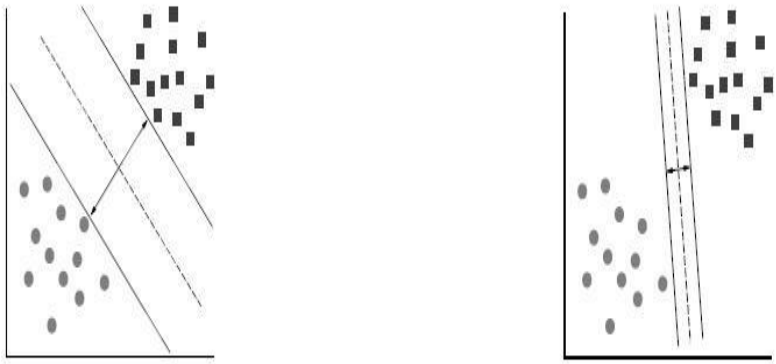

Figure 4.3: Support Vector Machines

belongs either to positive or negative class can be denoted by:

$$(x_1, y_1), \ldots, (x_u, y_u),\ x_j \in R^n,\ y_j \in \{+1, -1\}$$

Here, x_j is a feature vector of the j-th sample represented by an n dimensional vector; y_j is its class label. u is the number of the given training samples. SVM separates positive and negative examples by a hyperplane defined by:

$$w \cdot x + b = 0,\ w \in R^n, b \in R \tag{4.5}$$

where "$\cdot$" stands for the inner product. In general, a hyperplane is not unique (Cortes and Vapnik, 1995). The SVM determines the optimal hyperplane by maximizing the margin. The margin is the distance between negative examples and positive examples; the

distance between $w \cdot x + b = 1$ and $w \cdot x + b = -1$.

From Figure 4.3, we clearly see that the SVM on the left will generalize far better than that of the right as it has a optimally maximized margin between two classes of samples. The examples on $w \cdot x + b = \pm 1$ are called the Support Vectors which represent both positive or negative examples. The hyperplane must satisfy the following constraints:

$$y_i (w \cdot x_j + b) - 1 \geq 0$$

Hence, the size of the margin is $2/||w||$. In order to maximize the margin, we assume the following objective function:

$$\begin{aligned} Minimize_{w,b} J(w) &= \frac{1}{2}||w||^2 \\ s.t.\ y_j (w \cdot x_j + b) - 1 &\geq 0 \end{aligned} \tag{4.6}$$

By solving a quadratic programming problem, the decision function $f(x) = sgn(g(x))$ is derived, where

$$g(x) = \sum_{i=1}^{u} \lambda_i y_i x_i \cdot x + b \tag{4.7}$$

When examples are not linearly separable, the SVM algorithm allows for the use of slack variables (ξ_j) for all x_j to allow classification errors and the possibility to map examples to a (high-dimensional) feature space. These ξ_j give a misclassification error and

should satisfy the following inequalities (Kudo and Matsumoto, 2001):

$$y_i (w \cdot x_j + b) - (1 - \xi_j) \geq 0$$

Hence, we assume the following objective function to maximize the margin:

$$\begin{aligned} Minimize_{w,b,\xi}\, J(w,\xi) &= \frac{1}{2}||w||^2 + C\sum_{j=1}^{u}\xi_j \\ s.t.\ y_j (w \cdot x_j + b) - (1 - \xi_j) &\geq 0 \end{aligned} \tag{4.8}$$

Here, $||w||/2$ indicates the size of the margin, $\sum_{j=1}^{u}\xi_j$ indicates the penalty for misclassification, and C is the cost parameter that determines the trade-off for these two arguments. The decision function depends only on support vectors ($\lambda_i \neq 0$). Training examples, except for support vectors ($\lambda_i = 0$), have no influence on the decision function.

SVMs can handle non-linear decision surfaces with kernel function $K(x_i \cdot x)$. Therefore, the decision function can be rewritten as follows:

$$g(x) = \sum_{i=1}^{u}\lambda_i y_i K(x_i, x) + b \tag{4.9}$$

In this research, we use polynomial kernel functions, which have been found to be very effective in the study of other tasks in natural language processing (Joachims, 1998; Kudo and Matsumoto, 2001):

$$K(x,y) = (x \cdot y + 1)^d \tag{4.10}$$

4.6 Ensemble Methods

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a (weighted) vote of their predictions (Dietterich, 2000). The main strategy is to improve the overall performance by correcting mistakes of one classifier using the accurate output of others. Thus, ensembles are often much more accurate than the individual base models that make them up. Ensemble learning consists of two problems; *ensemble generation*: how does one generate the base models? and *ensemble integration*: how does one integrate the base models predictions to improve performance? Ensemble generation can be characterized as being *homogeneous* if each base learning model uses the same learning algorithm or *heterogeneous* if the base models can be built from a range of learning algorithms (Rooney et al., 2004). Many methods for constructing ensembles have been developed over the years which consider Bayesian voting, manipulation of the training examples, input features and output targets, injecting randomness and so on. The most general purpose homogeneous ensemble methods are Bagging, AdaBoost algorithm and Cross-Validation Committees (CVC) (Dietterich, 2000). Ensemble methods are successfully applied in text classification tasks (Silva and Ribeiro, 2006).

In this research, we use the Cross-Validation Committees (Parmanto, Munro, and Doyle, 1996) approach of constructing an homogeneous ensemble to inject differences into several SVM classifiers. We apply the supervised learning techniques: Support Vector Machines

(SVM), Hidden Markov Models (HMM), Conditional Random Fields (CRF), and Max-Ent (Maximum Entropy) to get individual predictions and then combine them to form a heterogeneous ensemble.

4.6.1 Homogeneous Ensemble

Ensemble generation for homogeneous learning is generally addressed by using different samples of the training data for each base model keeping the learning algorithm stable (Rooney et al., 2004). We use the CVC approach to make four different SVM classifiers.

Cross-Validation Committees (CVC) This is a training set sampling method where the strategy is to construct the training sets by leaving out disjoint subsets of the training data (Dietterich, 2000). For instance, the training set can be randomly divided into 4 disjoint subsets. Then 4 overlapping training sets can be built by dropping out a different one of these 4 subsets. As the same type of procedure is employed to construct training sets for 4-fold cross validation, so ensembles constructed in this manner is termed as *cross-validation committees* (Parmanto, Munro, and Doyle, 1996). An important issue in the CVC is the degree of data overlap between the replicates that is the different training subsets. The degree of overlap essentially depends on the number of replicates and the size of a removed fraction from the original sample.

1. Divide whole training data set D into $v-$fractions $d_1, \ldots, d_v$
2. Leave one fraction d_k and train classifier c_k with the rest of the data $(D - d_k)$
3. Build a committee from the classifiers using a simple averaging procedure.

Algorithm 1: Cross-Validation Committees (CVC) Method

The fraction of data overlap determines the trade-off between the individual classifier performance and error correlation between the classifiers. Lower correlation is often ob-

vious if the classifiers are trained with less overlapped data. The typical algorithm of the CVC approach (Parmanto, Munro, and Doyle, 1996) is presented in Algorithm 1.

4.6.2 Heterogeneous Ensemble

Heterogeneous ensemble is formed using the same training data set on different learning methods. We combine the individual decisions of the four classifiers: Support Vector Machines (SVM), Hidden Markov Models (HMM), Conditional Random Fields (CRF), and MaxEnt (Maximum Entropy) by taking a weighted voting and then the combined decision values are used to classify the unseen data set.

4.7 Summary

In this chapter, we discussed the theories of supervised machine learning techniques that we apply for the complex question answering task. In the next chapter, we will describe different evaluation techniques that can be used to judge all the systems.

Chapter 5

Evaluation Techniques

5.1 Introduction

In any natural language processing task, without systematic evaluation it is impossible to assess the quality of an NLP system and compare performance against other systems. Many NLP tasks such as parsing, named entity recognition, chunking and semantic role labeling etc. can be automatically evaluated using standard precision and recall measures. However, the evaluation of a summary is a very difficult task as there is no unique gold standard. For example, in automatic summarization there can be multiple possible summaries of the same source documents. Again, it is always hard to tell what makes a summary a good summary since this fact largely depends on who is the summarizer. For a summary to be a good summary, it must be comparable to an already defined good summary. In this chapter, we discuss the widely available summary evaluation techniques.

5.2 Manual Evaluation

In early 1960s, the evaluation of summaries was mainly done by humans (Edmundson, 1969). Methods for evaluating text summarization can be broadly classified into two categories (Jones and Galliers, 1996).

5.2.1 Intrinsic Methods

In intrinsic evaluation, humans judge the summarization quality based on the analyses of the summaries directly. This type of evaluation might involve user judgment of fluency of

summary coverage or similarity to an "ideal" summary. Measures of fluency can address language complexity, redundancy, coherence, preservation of different structured environments such as lists or tables, grammatical features etc.

User Evaluation

In DUC-2007, NIST manually evaluated the linguistic features of each submitted summary using a set of quality questions[1]. According to them, linguistic quality questions are targeted to assess how readable and fluent the summaries are, and they measure qualities of the summary that do not involve comparison with a model (human generated) summary or given topic. These questions require a certain readability property to be assessed on a five-point scale from "1" to "5", where "5" indicates that the summary is good with the respect to the quality under question, "1" indicates that the summary is bad with respect to the quality stated in the question, and "2" to "4" show the gradation in between. The quality of the summary is assessed only with respect to the property that is described in the specific category. The information content and responsiveness of the summary are measured separately in the "responsiveness" part of the evaluation.

Grammaticality The summary should have no datelines, system-internal formatting, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read.

Non-redundancy There should be no unnecessary repetition in the summary. Unnecessary repetition might take the form of whole sentences that are repeated, or repeated facts, or the repeated use of a noun or noun phrase (e.g., "Morris Dees") when a pronoun ("he")

[1] http://www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt

would suffice.

Referential clarity It should be easy to identify who or what the pronouns and noun phrases in the summary are referring to. If a person or other entity is mentioned, it should be clear what their role in the story is. So, a reference would be unclear if an entity is referenced but its identity or relation to the story remains unclear.

Focus The summary should have a focus. Sentences should only contain information that is related to the rest of the summary.

Structure and Coherence The summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to sentence to a coherent body of information about a topic.

Responsiveness This is measured primarily in terms of the amount of information present in the summary that actually helps to satisfy the information need expressed in the topic statement. The linguistic quality of the summary might play only an indirect role in this judgment, insofar as poor linguistic quality interferes with the expression of information and reduces the amount of information that is conveyed.

Pyramid Evaluation

The pyramid method is another manual evaluation technique for summarization evaluation and it was developed by Columbia University[2] in an attempt to address a key problem in summarization, namely the fact that different humans choose different content when writing summaries. The pyramid method addresses the problem by using multiple human

[2]http://www1.cs.columbia.edu/ becky/DUC2006/2006-pyramid-guidelines.html

summaries to create a gold-standard and by exploiting the frequency of information in the human summaries in order to assign importance to different facts. The pyramid approach tailors the evaluation to observed distributions of content over a pool of human summaries, rather than to human judgments of summaries. This method involves semantic matching of content units to which differential weights are assigned based on their frequency in a corpus of summaries that can lead to more stable, more informative scores, and hence to a meaningful content evaluation.

A pyramid is a model predicting the distribution of information content in summaries, as reflected in the summaries humans write. The pyramid model explicitly represents the overlapping content in a set of model human summaries, and indicates the frequency that models express each content unit. The resulting pyramid is used to evaluate the quality of information content in a new, distinct summary (a peer). For each topic, a weighted inventory of Summary Content Units (SCUs) i.e. a pyramid is created. The pyramid is reliable, predictive and diagnostic, and it constitutes a resource for investigating alternate realizations of the same meaning. An SCU is similar to a collection of paraphrases in that it groups together words and phrases from distinct summaries into a single set, based on shared content. The words selected from one summary to go into an SCU are referred to as a contributor of the SCU. The annotator must assign a label to the SCU that expresses the shared content. The label is a concise English sentence that states what the annotator views as the meaning of the content unit. Coincidentally, the SCU will have a weight corresponding to the number of model summaries that express the designated content. The SCU weight is automatically computed, based the number of summaries that contribute to it, so the annotator is not responsible for assigning weights. In DUC-2007 main task, modified pyramid scores are used that are very closely related to the original pyramid score, which equals the sum of the weights of the Summary Content Units (SCUs) that a peer summary matches, normalized by the weight of an ideally informative summary consisting

of the same number of contributors as the peer. However, the normalization factor for the modified score is the ideal weight of a summary which has the same number of contributors as the average of the model summaries in the associated pyramid.

5.2.2 Extrinsic Methods

If we determine the effect of summarization on some other task, that is termed as extrinsic evaluation. The usefulness of a summary can be examined with respect to some information needs such as finding documents from a large collection, routing documents, producing an effective report or presentation using a summary etc. It is also possible to judge the impact of a summarizer on the system in which it is embedded, for example, in a question answering system. The amount of work required to post-edit a summary output to make it more readable can be thought of as another measure to evaluate it.

5.3 Automatic Evaluation

Although manual evaluations provide essential feedback on the quality of system output, they are costly and time consuming to run. During system development, evaluations must be performed quickly and frequently and is thus impractical to elicit human judgments for development purposes. Due to this, researchers seek methods for automatically evaluating system output without any human input. The main challenge of automatic evaluation methods is the unavailability of an "ideal" summary for a direct comparison with system generated summary. The human summaries may be supplied by someone but there is no guarantee of these being perfect summaries since they might have considerable disadvantages. Hence, the judgment of a summary becomes increasingly difficult.

ROUGE *ROUGE*, which stands for "Recall-Oriented Understudy for Gisting Evaluation" (Lin, 2004), is an automatic summary evaluation toolkit that is widely accepted now-a-days. ROUGE is a collection of measures that determines the quality of a summary by comparing it to reference summaries created by humans. The measures count the number of overlapping units such as n-gram, word-sequences, and word-pairs between the system-generated summary to be evaluated and the ideal summaries created by humans. ROUGE measures considered in the evaluation are: ROUGE-N (N=1,2,3,4), ROUGE-L, ROUGE-W and ROUGE-S. We discuss the ROUGE similarity measures in section 3.2.

5.4 Our Approach

We evaluate all the system generated summaries with respect to the given abstract summaries by both manual and automatic evaluation methods. For manual evaluation, we follow the intrinsic approach and do both user and Pyramid evaluation according to the DUC-2007 guidelines.

5.5 Summary

In this chapter, we discuss the different summary evaluation techniques. Next chapter will focus on the implementation related issues in details.

Chapter 6

Implementation Details

6.1 Introduction

The complex question answering problem is a general one. One instance of it was the problem defined in the DUC-2007 main task. In this thesis, we consider this task to run our experiments. We accomplish the task by applying different supervised learning techniques. As supervised learning requires a huge amount of data in the training stage, we apply ROUGE similarity measure (Lin, 2004), Basic Element (BE) overlap (Hovy et al., 2006), syntactic similarity measure (Moschitti and Basili, 2006), semantic similarity measure (Moschitti et al., 2007), and Extended String Subsequence Kernel (ESSK) (Hirao et al., 2003) (discussed in Chapter 3) to automatically label the corpora of sentences and produce sufficient data for training. We feed these 5 types of labeled data into the learners of each of the supervised approaches: Support Vector Machines (SVM), Conditional Random Fields (CRF), Hidden Markov Models (HMM), and Maximum Entropy (MaxEnt). Then we extensively investigate the performance of the classifiers to label unseen sentences as summary or non-summary sentence. We also experiment with homogeneous and heterogeneous ensembles for the same task. This chapter discusses all the implementation issues.

6.2 Task Definition

Over the past three years, complex questions have been the focus of much attention in both the automatic Question Answering (QA) and Multi Document Summarization (MDS) communities. While most current complex QA evaluations (including the 2004 AQUAINT Re-

lationship QA Pilot[1], the 2005 Text Retrieval Conference (TREC) Relationship QA Task[2], and the TREC definition[3]) require systems to return unstructured lists of candidate answers in response to a complex question, recent MDS evaluations (including the 2005, 2006 and 2007 Document Understanding Conferences (DUC)[4]) have tasked systems with returning paragraph-length answers to complex questions that are responsive, relevant, and coherent. The DUC conference series is run by the National Institute of Standards and Technology (NIST) to further progress in summarization and enable researchers to participate in large-scale experiments.

The problem definition at DUC-2007 (now TAC[5]) was: *"Given a complex question (topic description) and a collection of relevant documents, the task is to synthesize a fluent, well-organized 250-word summary of the documents that answers the question(s) in the topic"*.

For example, given the topic description (from DUC-2007):

```
<topic>
    <num>D0703A</num>
       <title> steps toward introduction
       of the Euro </title>
       <narr>
        Describe steps taken and worldwide
        reaction prior to the introduction
        of the Euro on January 1, 1999.
        Include predictions and expectations
```

[1] http://trec.nist.gov/data/qa/add_QAresources/README.relationship.txt
[2] http://trec.nist.gov/data/qa/2005_qadata/qa.05.guidelines.html
[3] http://trec.nist.gov/overview.html
[4] http://duc.nist.gov/
[5] http://www.nist.gov/tac/

```
        reported in the press.
        </narr>
</topic>
```

and a collection of relevant documents, the task of the summarizer is to build a summary that answers the question(s) in the topic description. We consider this task and apply all four supervised approaches to generate topic-oriented 250-word extract summaries.

6.3 Corpus

The DUC-2006 and DUC-2007 document sets came from the AQUAINT corpus, which is comprised of newswire articles from the Associated Press and New York Times (1998-2000) and Xinhua News Agency (1996-2000). We use the DUC-2006 data to train all the systems and then produce extract summaries for the 25 topics of the DUC-2007 data according to the task description.

6.4 Data Processing

We clean up the raw data and extract information about the topics by deleting all the unnecessary tags. The sentences of each given document are tokenized by placing one sentence in each line. We do this to label the sentences (by +1 or -1 meaning summary or non-summary sentence) individually. We use OAK system (Sekine, 2002) for this purpose.

6.5 Feature Extraction

We represent each sentence of a document as a vector of feature-values. We divide the features into two major categories: the features which declare the importance of a sentence in a document and the features which measure the similarity between each sentence and the user query (Chali, Joty, and Hasan, 2009; Chali and Joty, 2008; Edmundson, 1969; Sekine and Nobata, 2001).

Importance Measures

Position of Sentences We give the score 1 to those sentences found within the first and the last 3 sentences of a document and assign score 0 to the rest, as the early and late sentences are considered important intuitively.

Length of Sentences If a sentence is longer, we can heuristically claim that it has a better chance of inclusion in the summary. We give the score 1 to a longer sentence and assign the score 0 otherwise. In this thesis, we considered a sentence as long if it has more than 11 words.

Title Match If we find a match such as exact word overlap, synonym overlap and hyponym overlap between the title and a sentence, we give it the score 1, otherwise 0.

Named Entity The score 1 is given to a sentence, which contains a certain Named Entity class among: PERSON, LOCATION, ORGANIZATION, GPE (Geo-Political Entity), FACILITY, DATE, MONEY, PERCENT, TIME. We use *OAK* System (Sekine, 2002), from New York University for Named Entity recognition.

Cue Word Match The probable relevance of a sentence is affected by the presence of pragmatic words such as “significant”, “impossible”, “in conclusion”, “finally” etc. We use a cue word list of 228 words. We give the score 1 to a sentence having any of the cue words and 0 otherwise.

Query-related Features

n-gram Overlap This is the recall between the query and the candidate sentence where n stands for the length of the n-gram$(n = 1,2,3,4)$.

LCS Given two sequences S_1 *and* S_2, the longest common subsequence (LCS) of S_1 *and* S_2 is a common subsequence with maximum length.

WLCS Weighted Longest Common Subsequence (WLCS) improves the basic LCS method to remember the length of consecutive matches encountered so far (Lin, 2004). We compute the WLCS-based F-measure between a query and a sentence.

Skip-Bigram Skip-bigram measures the overlap of skip-bigrams between a candidate sentence and a query sentence. Skip-bigram counts all in-order matching word pairs while LCS only counts one longest common subsequence.

Exact-word Overlap This is a measure that counts the number of words matching exactly between the candidate sentence and the query sentence.

Synonym Overlap This is the overlap between the list of synonyms of the important words extracted from the candidate sentence and the query related words. We use WordNet (Fellbaum, 1998) database for this purpose.

Hypernym/Hyponym Overlap It is the overlap between the list of hypernyms and hyponyms (up to level 2 in WordNet) of the nouns extracted from the sentence and the query related words.

Gloss Overlap Our systems extract the glossary entry for the proper nouns from WordNet. Gloss overlap is the overlap between the list of important words that are extracted from the glossary definition of the nouns in the candidate sentence and the query related words.

Syntactic Feature The syntactic similarity between the *query* and the *sentence* is calculated after parsing them into syntactic trees using a parser such as (Charniak, 1999) and finding the similarity between the two trees using the *tree kernel* (Collins and Duffy, 2001).

Basic Element (BE) Overlap We extract BEs (Discussed in Section 3.3) for the sentences in the document collection. Then we filter those BEs by checking whether they contain any word which is a *query word* or a *query related word* and get the BE overlap score (Hovy, Lin, and Zhou, 2005).

6.6 Experimental Setup

6.6.1 Training and Testing Data Preparation

We use the five automatic annotation methods to label each sentence of the 50 document sets of DUC-2006 to produce five different versions of training data for feeding the SVM, HMM, CRF and MaxEnt learners. We choose the top 30% sentences (based on the scores assigned by an annotation scheme) of a document set to have the label $+1$ and the rest to

have -1. Unlabeled sentences of 25 document sets of DUC-2007 data are used for the testing purpose.

In another experiment, to check whether a balanced set of training data improves the performance of the supervised systems or not, we obtain a training data set by annotating (using only ROUGE similarity measures(Lin, 2004)) 50% sentences of each document set as positive and the rest as negative. The ensemble experiments are also performed using this balanced set of training data.

Typically, the training data includes a collection of sentences where each sentence is represented as a combination of a feature vector and corresponding class label ($+1$ or -1). On the other hand, testing data is comprised of a set of sentences that are represented as feature vectors. The organization of training and testing data depends on the input format of the package that is used for a particular supervised system.

6.6.2 Package Settings

SVM

We use the second order polynomial kernel for the ROUGE and ESSK labeled training sets. For the BE, syntactic and semantic labeled training sets, the third order polynomial kernel is used. The third order polynomial kernel is also used when we do experiments with the balanced training data. The use of each kernel is based on the accuracy we achieved during training.

To allow some flexibility in separating the classes, SVM models have a cost parameter, C, that controls the trade off between allowing training errors and forcing rigid margins. It creates a soft margin that permits some misclassifications. Increasing the value of C increases the cost of misclassifying points and forces the creation of a more accurate model

that may not generalize well. We apply 3-fold cross validation with randomized local-grid search (Hsu, Chang, and Lin, 2008) for estimating the value of the trade off parameter C. Intuitively, we try the value of C in 2^i, where $i \in \{-5, -4, \cdots, 4, 5\}$ and set C as the best performed value of 0.125 for the second order polynomial kernel. We keep the default value for the third order polynomial kernel. We use the SVM^{light} (Joachims, 1999) package[6] for training and testing in this work. SVM^{light} is an implementation of Support Vector Machine (Cortes and Vapnik, 1995) for the problems of pattern recognition, regression, and learning a ranking function. The optimization algorithms used here have scalable memory requirements and can handle problems with many thousands of support vectors efficiently. This software also provides methods for assessing the generalization performance efficiently. It includes two efficient estimation methods for both error rate and precision/recall. SVM^{light} consists of a learning module and a classification module. The learning module takes an input file containing the feature values with corresponding labels and produces a model file. The classification module is used to apply the learned model to new examples.

HMM

We apply Maximum Likelihood Estimation[7] (MLE) technique by frequency counts with add-one smoothing[8] to estimate the three HMM parameters: initial state probabilities, transition probabilities and emission probabilities. We use Dr. Dekang Lin's HMM package[9] to generate the most probable label sequence given the model parameters and the observation

[6] http://svmlight.joachims.org/

[7] The idea behind maximum likelihood parameter estimation is to determine the parameters that maximize the probability (likelihood) of the sample data. From a statistical point of view, the method of maximum likelihood is considered to be more robust (with some exceptions) and yields estimators with good statistical properties.

[8] This method merely adds one to each count to modify the maximum likelihood estimates for computing the probabilities, focusing on the events that are incorrectly assumed to have zero probability.

[9] http://www.cs.ualberta.ca/~lindek/hmm.htm

sequence (unlabeled DUC-2007 test data).

CRF

We use MALLET-0.4 NLP toolkit[10] (McCallum, 2002) to implement the CRF. We formulate our problem in terms of MALLET's SimpleTagger[11] class, which is a command line interface to the MALLET CRF class. We modify the SimpleTagger class in order to include the provision for producing corresponding posterior probabilities of the predicted labels which are used later for ranking sentences.

MaxEnt

We build the MaxEnt system using Dr. Dekang Lin's MaxEnt package[12]. To define the exponential prior of the λ values[13] in MaxEnt models, an extra parameter α is used in the package during training. We keep the value of α as default.

[10]MALLET is a Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text. MALLET includes sophisticated tools for document classification, efficient routines for converting text to "features", a wide variety of algorithms (including Nave Bayes, Maximum Entropy, and Decision Trees), and code for evaluating classifier performance using several commonly used metrics. In addition to classification, MALLET includes tools for sequence tagging for applications such as named-entity extraction from text. Algorithms include Hidden Markov Models, Maximum Entropy Markov Models, and Conditional Random Fields. These methods are implemented in an extensible system for finite state transducers.

[11]http://mallet.cs.umass.edu/index.php/

[12]http://www.cs.ualberta.ca/~lindek/downloads.htm

[13]λ is the associated weight for each feature, which is learned by the MaxEnt model using numerical optimization techniques.

Ensemble Experiments

Homogeneous Ensemble To build a homogeneous ensemble, we generate 4 different SVM models in the following way. We divide the training data set (DUC-2006 data) into 4 equal-sized groups. According to the Cross-Validation Committees (CVC) algorithm (Parmanto, Munro, and Doyle, 1996) (discussed in Section 4.6.1), each time we keep 25% of the data aside and use the remaining 75% data for training. Next, we present the test data (DUC-2007 data) before each of the generated SVM models which produces individual predictions (decision scores along with a label $+1$ or -1) to those unseen data. The decision scores are the normalized distance from the separating hyperplane[14] to each sample. Then, we create the SVM ensemble by combining the predictions by simple weighted averaging. We increment a particular classifier's decision value by 1 (giving more weight) if it predicts a sentence as positive and decrement by 1 (imposing penalty), if the case is opposite. The resulting prediction values are used later for ranking the sentences. During training steps, we use the third-order polynomial kernel for the SVM keeping the default value of the trade-off parameter C. We perform the training experiments in WestGrid[15], which operates a high performance computing (HPC) collaboration and visualization infrastructure across western Canada. We use the *Cortex* cluster which is comprised of some shared-memory computers for large serial jobs or demanding parallel jobs.

Heterogeneous Ensemble Differences among the classifiers can be realized by using separate training samples with the same learning method (Qi and Huang, 2007). We experiment with an ensemble method that uses the same training set on different learning methods. Thus, we consider it as a heterogeneous ensemble that joins the above four classifiers (SVM, HMM, CRF and MaxEnt) which are somehow different in accomplishing

[14] A Support Vector Machine (SVM) performs classification by constructing an N-dimensional hyperplane that optimally separates the data into two categories.

[15] http://westgrid.ca/

the classification task. We combine the individual decisions of the classifiers by taking a weighted voting. Then, the combined decision values are used to label the unseen data set. We impose a positive weight (ranging from 1 to 5 depending on the individual classifier's performance, more weight if it is declared positive by a better performer) to each positively classified sentence. We take no action for the negatively classified sentences so that they can fall back during ranking.

6.6.3 Sentence Ranking

Extract summary generation can be thought of as searching for important sentences in the documents, which can be dealt with as a two-class problem. However, the proportion of important sentences in training data will differ from that in test data. The number of important sentences in a document is determined by a summarization rate or word limit which is given at run-time. In the Multi-Document Summarization task at DUC-2007, the word limit was 250 words. A simple solution to this problem is to rank sentences in a document, then select the top N sentences.

In SVM systems, we use $g(x)$, the normalized distance from the hyperplane to each sample point, x to rank the sentences. Then, we choose N sentences until the summary length (250 words for DUC-2007) is reached. For HMM systems, we use Maximal Marginal Relevance (MMR) based method to rank the sentences (Carbonell, Geng, and Goldstein, 1997). According to MMR, we choose a sentence for inclusion in summary such that it is maximally similar to the document and dissimilar to the already-selected sentences. In CRF systems, we generate posterior probabilities corresponding to each predicted label in the label sequence to measure the confidence of each sentence for summary inclusion. Similarly, for MaxEnt, the corresponding probability values of the predicted labels are used to rank the sentences.

6.7 Summary

In this chapter, we presented the implementation issues related to the systems that we used for the complex question answering task. Next chapter will show the results of all the experiments followed by relevant discussions.

Chapter 7

Results and Analyses

7.1 Introduction

In DUC-2007, each topic and its document cluster were given to 4 different NIST assessors, including the developer of the topic. The assessor created a 250-word summary of the document cluster that satisfies the information need expressed in the topic statement. These multiple "reference summaries" are used in the evaluation of our summary content. In this chapter, we present the automatic evaluation and manual evaluation results of all our systems.

7.2 Automatic Evaluation Results

We evaluate the system generated summaries using the automatic evaluation toolkit ROUGE (Lin, 2004) which has been widely adopted by DUC (Now TAC[1]). The available ROUGE measures are: ROUGE-N (N=1,2,3,4), ROUGE-L, ROUGE-W and ROUGE-S. ROUGE parameters were set as that of DUC-2007 evaluation setup.

Precision (P) and Recall (R) are the widely used evaluation measures computed based on the number of units (i.e. sentences, words, etc) common to both system-generated and reference summaries. F-measure, another important measure in NLP, combines precision and recall into a single measure of overall performance. We consider these widely used evaluation measures Precision (P), Recall (R) and F-measure for our evaluation task. We report the three widely adopted ROUGE metrics in the results: ROUGE-1 (unigram), ROUGE-2 (bigram) and ROUGE-SU (skip bi-gram) because these have never been shown

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

to not correlate with the human judgement. All the ROUGE measures are calculated by running ROUGE-1.5.5 with stemming but no removal of stopwords.

ROUGE run-time parameters:

ROUGE-1.5.5.pl -2 -1 -u -r 1000 -t 0 -n 4 -w 1.2 -m -l 250 -a

We show 95% confidence interval of the evaluation metric, ROUGE-SU for all systems to report significance for doing meaningful comparison. We use the ROUGE tool for this purpose. ROUGE uses a randomized method named bootstrap resampling to compute the confidence interval. Bootstrap resampling has a long tradition in the field of statistics (Efron and Tibshirani, 1994). The assumption here is that, estimating the confidence interval from a large number of test sets with n test samples drawn from a set of n test samples with replacement is as good as estimating the confidence interval for the test sets of size n from a large number of test sets with n test samples drawn from an infinite set of test samples. The benefit of this assumption is that we only need to consider n samples. We use 1000 sampling points in the bootstrap resampling.

7.2.1 Impact of Automatic Annotation Techniques

The main goal of this research is to study the impact of different automatic annotation techniques on the performance of the supervised approaches to the complex question answering task. To accomplish this, we generated summaries for 25 topics of DUC-2007 data by each of our four supervised systems: SVM, HMM, CRF and MaxEnt with each system trained using five different automatic labeling methods.

Table 7.1 to Table 7.4 show the ROUGE F-measures for SVM, HMM, CRF and MaxEnt systems, respectively. In the first column, **ROUGE**, **BE**, **Synt** (Syntactic), **Sem** (Semantic) and **ESSK** stand for the annotation scheme used. We highlight the top F-scores in each

table to indicate significance at a glance.

Annotation	ROUGE-1	ROUGE-2	ROUGE-SU
ROUGE	0.3838	0.0780	0.1432
BE	0.3855	**0.0890**	0.1470
Synt	0.3755	0.0757	0.1363
Sem	**0.3905**	0.0867	**0.1475**
ESSK	0.3738	0.0758	0.1358

Table 7.1: ROUGE F-measures for SVM

Annotation	ROUGE-1	ROUGE-2	ROUGE-SU
ROUGE	0.3940	0.0916	0.1509
BE	0.3684	0.0879	0.1377
Synt	0.3689	0.0863	0.1378
Sem	0.3387	0.0797	0.1207
ESSK	**0.3959**	**0.0931**	**0.1517**

Table 7.2: ROUGE F-measures for HMM

Annotation	ROUGE-1	ROUGE-2	ROUGE-SU
ROUGE	0.3748	0.0776	0.1346
BE	0.3619	0.0611	0.1241
Synt	0.3631	0.0688	0.1265
Sem	0.3743	**0.0777**	0.1332
ESSK	**0.3813**	0.0746	**0.1385**

Table 7.3: ROUGE F-measures for CRF

In Table 7.1, we can see that the *ESSK* labeled SVM system is having the poorest ROUGE-1 score whereas the *Sem* labeled system performs best. The other annotation methods' impact is almost similar here in terms of ROUGE-1. Analyzing ROUGE-2 scores, we find that the *BE* performs the best for SVM, on the other hand, *Sem* achieves

Annotation	ROUGE-1	ROUGE-2	ROUGE-SU
ROUGE	0.3938	0.0871	0.1490
BE	0.3739	0.0703	0.1320
Synt	0.3942	0.0838	0.1502
Sem	0.3876	0.0834	0.1454
ESSK	**0.4006**	**0.0923**	**0.1554**

Table 7.4: ROUGE F-measures for MaxEnt

top ROUGE-SU score. As for two measures *Sem* annotation is performing the best, we can typically conclude that *Sem* annotation is the most suitable method for the SVM system.

Similarly, analyzing Table 7.2 yields the fact that *ESSK* works best for HMM and *Sem* labeling does worst for all ROUGE scores. *Synt* and *BE* labeled HMMs perform almost similar whereas *ROUGE* labeled system is pretty close to that of *ESSK*.

In Table 7.3, we see that the CRF performs best with the *ESSK* annotated data in terms of ROUGE -1 and ROUGE-SU scores and *Sem* has the highest ROUGE-2 score. But *BE* and *Synt* labeling work bad for CRF whereas the *ROUGE* labeling performs close to *ESSK*. From this table, we can typically conclude that *ESSK* annotation is the best method for the CRF system.

From Table 7.4, we find that *ESSK* works best for MaxEnt and *BE* labeling is the worst for all ROUGE scores. We can also see that *ROUGE*, *Synt* and *Sem* labeled MaxEnt systems perform almost similar.

So, after analyzing the results of Table 7.1 we can come to a conclusion that SVM system performs best if training data uses semantic annotation scheme. Similarly, analysis on Tables 7.2 to 7.4 reveals that *ESSK* works best for HMM, CRF and MaxEnt systems.

Figure 7.1 shows the ROUGE F-measures for SVM, HMM, CRF and MaxEnt systems. The X-axis containing **ROUGE**, **BE**, **Synt** (Syntactic), **Sem** (Semantic), and **ESSK** stands for the annotation scheme used. The Y-axis shows the ROUGE-1 scores at the top,

ROUGE-2 scores at the bottom and ROUGE-SU scores in the middle. The supervised systems are distinguished by the line style used in the figure.

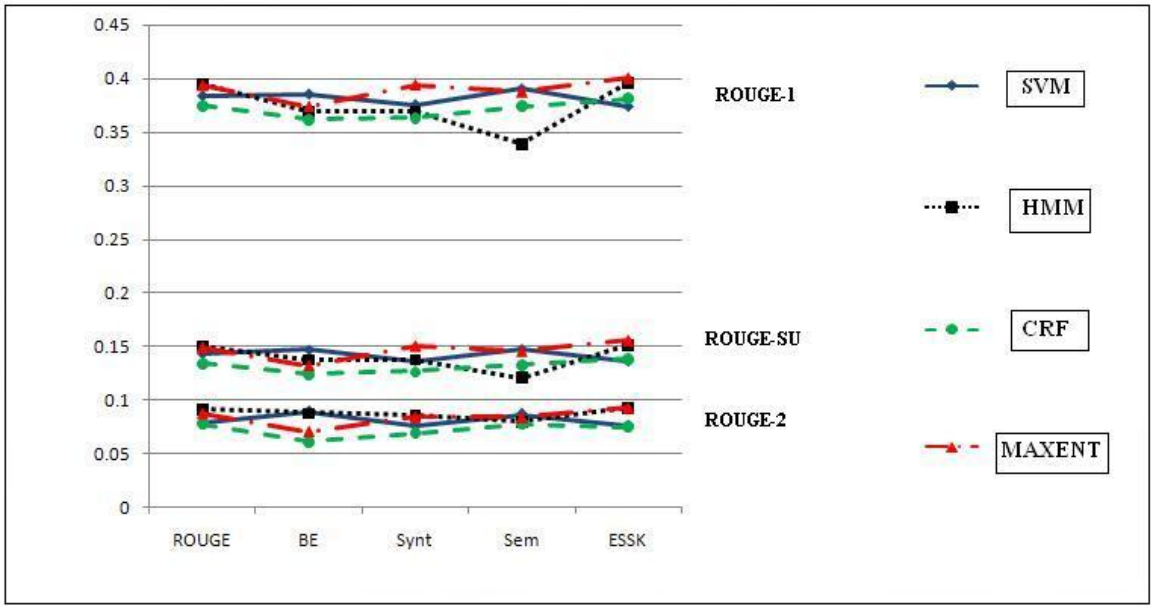

Figure 7.1: ROUGE F-scores for different supervised systems

For a direct comparison, in Table 7.5 we show average ROUGE F-Scores of one baseline system and four supervised approaches in terms of their best suited annotation method used. The baseline system generates summaries by returning all the leading sentences (up to 250 words) in the $\langle TEXT \rangle$ field of the most recent document(s). Table 7.5 shows that all the supervised systems typically outperform the baseline system with their best annotation method applied and the MaxEnt system performs best with SVM, HMM and CRF to follow.

From another angle of analysis, if we average all the corresponding ROUGE F - scores of the SVM, HMM, CRF and MaxEnt systems, we can clearly show the general impact of different annotation techniques on all the supervised approaches cumulatively in Table 7.6. Here, we find ESSK as the most effective annotation strategy. From Table 7.6, we can also infer that ROUGE is somewhat respectable as labeling method whereas BE, Syntactic and

Semantic techniques perform almost similar for the task of labeling.

System	ROUGE-1	ROUGE-2	ROUGE-SU
Baseline	0.3347	0.0640	0.1127
SVM(Sem)	0.3905	0.0867	0.1475
HMM(ESSK)	0.3959	0.0931	0.1517
CRF(ESSK)	0.3813	0.0746	0.1385
MaxEnt(ESSK)	0.4006	0.0923	0.1554

Table 7.5: F-measures of supervised systems (Comparison)

Annotation	ROUGE-1	ROUGE-2	ROUGE-SU
ROUGE	0.3866	0.0835	0.1444
BE	0.3724	0.0770	0.1352
Syntactic	0.3754	0.0787	0.1377
Semantic	0.3728	0.0819	0.1367
ESSK	0.3879	0.0840	0.1454

Table 7.6: General impact of annotation techniques

In Tables 7.7 to 7.10, we show the 95% confidence intervals of the F-measures for ROUGE - SU for SVM, HMM, CRF and MaxEnt systems respectively to meaningfully compare the impact of annotation methods.

Annotation	ROUGE-SU
ROUGE	0.130792 - 0.158095
BE	0.131675 - 0.162274
Synt	0.123773 - 0.151050
Sem	0.137251 - 0.158756
ESSK	0.121636 - 0.152182

Table 7.7: 95% confidence intervals for SVM

Analyzing the confidence intervals from Table 7.7 to Table 7.10, it is obvious that *Sem*

Annotation	ROUGE-SU
ROUGE	0.136643 - 0.163567
BE	0.117484 - 0.154962
Synt	0.117578 - 0.154498
Sem	0.097598 - 0.142473
ESSK	0.138965 - 0.163144

Table 7.8: 95% confidence intervals for HMM

Annotation	ROUGE-SU
ROUGE	0.123273 - 0.146486
BE	0.113709 - 0.135212
Synt	0.116490 - 0.137028
Sem	0.120812 - 0.144681
ESSK	0.126372 - 0.150955

Table 7.9: 95% confidence intervals for CRF

Annotation	ROUGE-SU
ROUGE	0.136783 - 0.161889
BE	0.119786 - 0.143856
Synt	0.136549 - 0.162500
Sem	0.132788 - 0.158256
ESSK	0.142045 - 0.167066

Table 7.10: 95% confidence intervals for MaxEnt

annotation is performing better than other methods for SVM system and *ESSK* is having the best scores for HMM, CRF and MaxEnt systems.

7.2.2 Balanced/Unbalanced Training Data

We perform another experiment by feeding a balanced set of training data (annotating 50% sentences as positive and the rest as negative by ROUGE similarity measure) into the learners of the supervised systems. Tables 7.11 to 7.13 present the ROUGE-F score comparisons of the supervised systems in terms of balanced and unbalanced (30% positive samples) training data. Analyses on these tables show that supervised systems perform well to confront the complex question answering task if they are trained on a data set where positive samples are in less numbers. This is because we always pick up a very small number of sentences to be included in the target summary.

Positive Samples	ROUGE-1	ROUGE-2	ROUGE-SU
30%	0.3838	0.0780	0.1432
50%	0.3708	0.0672	0.1328

Table 7.11: SVM comparisons based on balanced/unbalanced training data

Positive Samples	ROUGE-1	ROUGE-2	ROUGE-SU
30%	0.3940	0.0916	0.1509
50%	0.3945	0.0898	0.1499

Table 7.12: HMM comparisons based on balanced/unbalanced training data

Positive Samples	ROUGE-1	ROUGE-2	ROUGE-SU
30%	0.3748	0.0776	0.1346
50%	0.3725	0.0742	0.1329

Table 7.13: CRF comparisons based on balanced/unbalanced training data

7.2.3 Ensemble Experiments

For our ensemble experiments, we use the balanced training data set. We use the ROUGE similarity measure-based labeling data here. We performed this experiment early in this research, so we chose the balanced training data and ROUGE based labeling data since they were only available at that time.

Homogeneous Ensemble

We employ a SVM-based ensemble approach using the CVC algorithm. In Table 7.14, we present the ROUGE scores of the SVM ensemble system in terms of Precision, Recall and F-scores. Similarly, Table 7.15 shows the ROUGE scores of the single SVM system. The F-scores for the single SVM system, the baseline system and the SVM ensemble system are shown in Table 7.16. The single SVM system is trained on the full data set of DUC-2006. The approach of the baseline system is to select the lead sentences (up to 250 words) from a document set for each topic. Table 7.16 clearly suggests that the SVM ensemble system outperforms the baseline system with a high margin for all of the ROUGE measures. It also outperforms the single SVM system by a meaningful margin.

Figure 7.2 shows a clear view of how the SVM-based homogeneous ensemble performs better than the single SVM system and the baseline system.

In table 7.17, We report 95% confidence intervals of the F-measures for the single

Measures	ROUGE-1	ROUGE-2	ROUGE-SU
Precision	0.4081	0.0860	0.1621
Recall	0.3705	0.0781	0.1334
F-score	0.3883	0.0819	0.1463

Table 7.14: ROUGE measures for SVM ensemble

Measures	ROUGE-1	ROUGE-2	ROUGE-SU
Precision	0.3902	0.0707	0.1477
Recall	0.3534	0.0641	0.1209
F-score	0.3708	0.0672	0.1329

Table 7.15: ROUGE measures for single SVM

Systems	ROUGE-1	ROUGE-2	ROUGE-SU
Base	0.3347	0.0649	0.1127
Single	0.3708	0.0672	0.1329
Ensemble	0.3883	0.0819	0.1463

Table 7.16: Homogeneous ensemble comparison

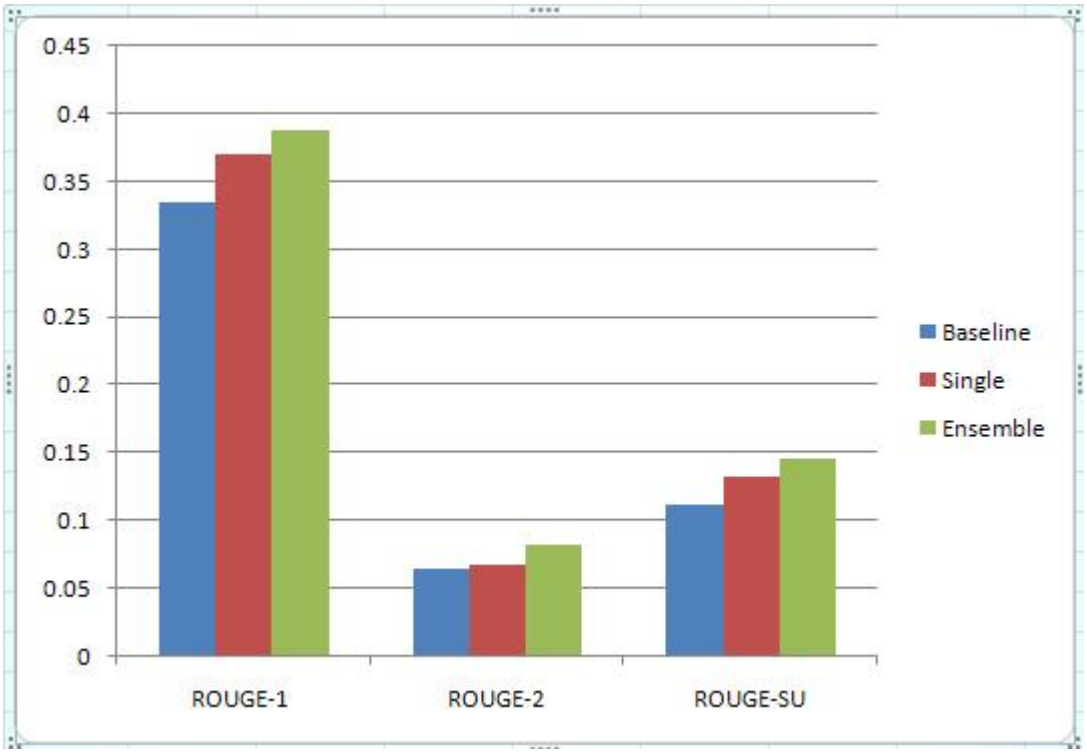

Figure 7.2: SVM-based ensemble beating single SVM

SVM system, the baseline system and the SVM ensemble system to show significance for meaningful comparison. We can see from table 7.17 that the ensemble system performs better than the single SVM and the baseline system.

Systems	ROUGE-1	ROUGE-2	ROUGE-SU
Baseline	0.326680 - 0.342330	0.060870 - 0.068840	0.108470 - 0.116720
Single	0.355833 - 0.386524	0.057032 - 0.078794	0.121819 - 0.144470
Ensemble	0.370439 - 0.406841	0.068727 - 0.094480	0.133385 - 0.159090

Table 7.17: 95% confidence intervals for different systems

Systems	ROUGE-1	ROUGE-2	ROUGE-SU
Baseline	0.3347	0.0649	0.1127
Ensemble	0.3950	0.0885	0.1530
HMM	0.3945	0.0898	0.1499
MaxEnt	0.3938	0.0871	0.1490
CRF	0.3725	0.0742	0.1329
SVM	0.3708	0.0672	0.1328

Table 7.18: Heterogeneous ensemble comparison

Heterogeneous Ensemble

We experiment with an ensemble based approach combining the individual decisions of the four classifiers: SVM, CRF, HMM and MaxEnt. Table 7.18 shows the ROUGE F-measures for SVM, HMM, CRF, MaxEnt, ensemble and the baseline system.

Table 7.18 clearly suggests that the most of the supervised systems outperform the baseline system by a high margin and the ensemble system is typically the best performer. This is because the individual classifier decisions are combined together to judge the sentence labels correctly. Comparison of the four supervised methods individually reveals that HMM is performing the best and MaxEnt, CRF and SVM are the next in the performance ranking, respectively. The reason for this is that HMM treats the task as a sequence labeling problem and the scores are improved further as we use the MMR method for sentence ranking, which has proven to be an effective way of reducing the redundancy. We can also understand that the MaxEnt method performs better than SVM, concluding that high-order kernels may not be suited well for the problem domain.

Figure 7.3 portrays the comparison of all the supervised systems. We find that the heterogeneous ensemble outperforms its individual counterparts, and HMM and MaxEnt perform close to it.

In table 7.19, We show the 95% confidence intervals of the F-measures of ROUGE-1

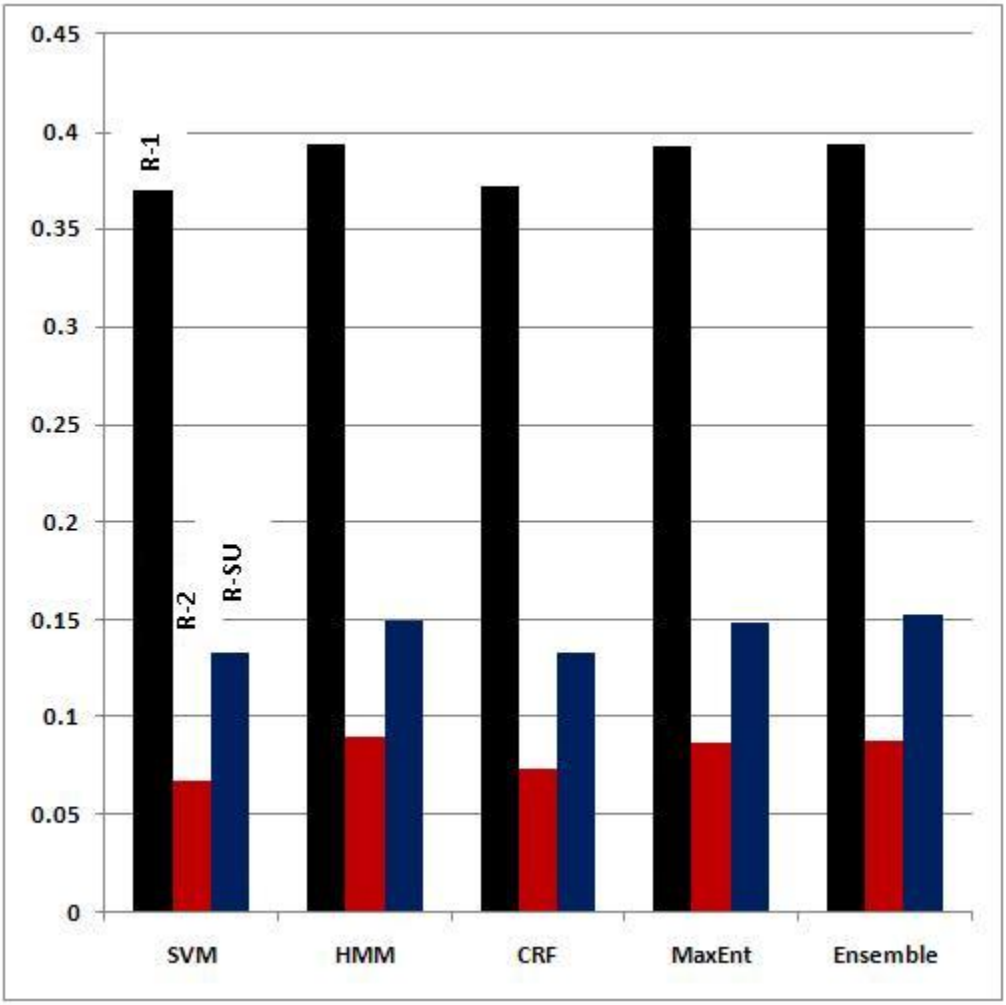

Figure 7.3: Comparison of supervised systems

and ROUGE-SU for the baseline system and our supervised systems.

Systems	ROUGE-1	ROUGE-SU
Baseline	0.3266 - 0.3423	0.1084 - 0.1167
Ensemble	0.3739 - 0.4127	0.1386 - 0.1663
HMM	0.3745 - 0.4107	0.1363 - 0.1613
MaxEnt	0.3763 - 0.4109	0.1367 - 0.1618
CRF	0.3553 - 0.3892	0.1205 - 0.1457
SVM	0.3558 - 0.3865	0.1217 - 0.1444

Table 7.19: 95% confidence intervals

If we analyze table 7.19, we find that all the supervised systems perform significantly better than the baseline system. On the other hand, for the ensemble system, HMM, and MaxEnt, we get a high overlap in terms of all the ROUGE measures.

7.3 Manual Evaluation Results

For a sample of 46 summaries[2] drawn from the generated summaries of our different systems, we conduct an extensive manual evaluation in order to analyze the effectiveness of our approaches. The manual evaluation is comprised of a Pyramid-based evaluation of contents and a user evaluation to obtain the assessment of linguistic quality and overall responsiveness.

7.3.1 User Evaluation

Two university graduate students judged the summaries for linguistic quality and overall responsiveness according to the DUC-2007 evaluation guidelines[3]. The given score is an

[2]Randomly, we choose 2 summaries for each of these systems.

[3]http://www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt

integer between 1 (very poor) and 5 (very good) and is guided by consideration of the following factors: 1. Grammaticality, 2. Non-redundancy, 3. Referential clarity, 4. Focus, and 5. Structure and Coherence. They also assigned a content responsiveness score to each of the automatic summaries. The content score is an integer between 1 (very poor) and 5 (very good) and is based on the amount of information in the summary that helps to satisfy the information need expressed in the topic narrative. Tables 7.20 to 7.24 present the average linguistic quality and overall responsive scores of all our systems. The same baseline system scores are given for meaningful comparison. Analysis on these tables indicates that the user evaluation results are not that much consistent with the automatic evaluation results. Since we conducted the user evaluation on a small set of sample summaries, we restrict ourselves to infer anything from these results.

Systems	Linguistic Quality	Overall Responsiveness
Baseline	4.24	1.80
ROUGE	4.40	3.00
BE	3.90	4.00
Synt	4.20	3.00
Sem	4.10	3.00
ESSK	4.40	4.00

Table 7.20: Linguistic quality and responsive scores for SVM

Systems	Linguistic Quality	Overall Responsiveness
Baseline	4.24	1.80
ROUGE	4.50	4.00
BE	4.20	3.50
Synt	3.70	3.00
Sem	4.40	4.00
ESSK	4.70	4.00

Table 7.21: Linguistic quality and responsive scores for CRF

Systems	Linguistic Quality	Overall Responsiveness
Baseline	4.24	1.80
ROUGE	4.30	3.50
BE	4.40	3.50
Synt	4.50	3.50
Sem	4.00	3.00
ESSK	4.10	3.50

Table 7.22: Linguistic quality and responsive scores for HMM

Systems	Linguistic Quality	Overall Responsiveness
Baseline	4.24	1.80
ROUGE	4.40	3.50
BE	4.30	3.50
Synt	4.10	3.00
Sem	4.40	3.50
ESSK	4.20	3.50

Table 7.23: Linguistic quality and responsive scores for MaxEnt

Systems	Linguistic Quality	Overall Responsiveness
Baseline	4.24	1.80
Single SVM	4.10	3.00
Homogeneous	3.90	3.00
Heterogeneous	4.50	3.00

Table 7.24: Linguistic quality and responsive scores for ensemble systems

7.3.2 Pyramid Evaluation

In the DUC 2007 main task, 23 topics were selected for the optional community-based pyramid evaluation. Volunteers from 16 different sites created pyramids and annotated the peer summaries for the DUC main task using the given guidelines[4]. Eight sites among them created the pyramids. We used these pyramids to annotate our peer summaries to compute the modified pyramid scores[5]. We used the *DUCView.jar*[6] annotation tool for this purpose. Tables 7.25 to 7.29 show the modified pyramid scores of all our systems. A baseline system score is also reported. The peer summaries of the baseline system are generated by returning all the leading sentences (up to 250 words) in the $\langle TEXT \rangle$ field of the most recent document(s). Again, we do not conclude anything from the pyramid evaluation results since the evaluation was limited to a small set of peer summaries.

Systems	Modified Pyramid Scores
Baseline	0.13874
ROUGE	0.43415
BE	0.44255
Synt	0.45675
Sem	0.47500
ESSK	0.43670

Table 7.25: Modified pyramid scores for SVM systems

[4]http://www1.cs.columbia.edu/ becky/DUC2006/2006-pyramid-guidelines.html

[5]This equals the sum of the weights of the Summary Content Units (SCUs) that a peer summary matches, normalized by the weight of an ideally informative summary consisting of the same number of contributors as the peer.

[6]*http://www1.cs.columbia.edu/ ani/DUC2005/Tool.html*

Systems	Modified Pyramid Scores
Baseline	0.13874
ROUGE	0.43530
BE	0.51075
Synt	0.37500
Sem	0.41190
ESSK	0.47505

Table 7.26: Modified pyramid scores for CRF systems

Systems	Modified Pyramid Scores
Baseline	0.13874
ROUGE	0.39915
BE	0.37500
Synt	0.48215
Sem	0.45270
ESSK	0.41825

Table 7.27: Modified pyramid scores for HMM systems

Systems	Modified Pyramid Scores
Baseline	0.13874
ROUGE	0.60665
BE	0.55675
Synt	0.50840
Sem	0.55995
ESSK	0.34740

Table 7.28: Modified pyramid scores for MaxEnt systems

Systems	Modified Pyramid Scores
Baseline	0.13874
Single SVM	0.35270
Homogeneous	0.34845
Heterogeneous	0.53000

Table 7.29: Modified pyramid scores for ensemble systems

7.4 Summary

In this chapter, we showed the automatic evaluation results and detailed analyses of all the experiments. We found that, based on the annotation scheme used, performance of the supervised systems varied. Typically, we concluded that Semantic annotation works well for the SVM system and ESSK does best for the HMM, CRF and MaxEnt systems. From another experiment, we inferred that systems trained with an unbalanced data set where positive samples are less in proportion often outperforms the systems that are trained with a balanced data set. Our ensemble experiments showed that both homogeneous and heterogeneous ensembles are performing better than their individual counterparts. We also showed the manual evaluation results with the intention to give meaningful comparisons, but we could not infer anything from these results due to a small sample size. In the coming chapter, we will conclude the thesis by providing some future directions of this research.

Chapter 8

Conclusions and Future Directions

8.1 Main Findings

Our main concern in this thesis is to address the issue of answering complex questions by using an extractive multi-document summarization approach in a supervised framework. We have investigated the broad spectrum of automatic text summarization. Then, we focus on applying different automatic annotation techniques and supervised methods to confront the problem. The following is a summary of the core findings and contributions of our work:

1. We solved the automatic annotation problem by applying five different sentence similarity measurement techniques: ROUGE similarity measure, Basic Element (BE) overlap, syntactic similarity measure, semantic similarity measure, and Extended String Subsequence Kernel (ESSK). In this manner we generated five different versions of labeled data that were used for training the supervised systems.

2. We formulated the complex question answering problem in terms of four different supervised machine learning techniques: Support Vector Machines (SVM), Hidden Markov Models (HMM), Conditional Random Fields (CRF), and Maximum Entropy (MaxEnt).

3. We conducted an extensive experimental analysis to show the impact of five automatic annotation methods on the performance of the four chosen supervised machine learning techniques. We evaluated our systems automatically using ROUGE and reported the significance of our results through 95% confidence intervals. Experimental results showed that *Sem* annotation is the best for SVM whereas *ESSK* works

well for HMM, CRF and MaxEnt systems. We also presented the manual evaluation results to compare our systems meaningfully.

4. We also assessed system performance by feeding balanced and unbalanced data during the learning phase. From this experiment we inferred that systems trained with an unbalanced data set where positive samples are less in proportion often outperforms the systems that are trained with a balanced data set.

5. We experimented with two supervised ensemble based approaches as well. The homogeneous ensemble was made with four different SVM classifiers whereas the heterogeneous ensemble combined the decisions of the four classifiers: SVM, CRF, HMM, and MaxEnt. Our experiments showed that both homogeneous and heterogeneous ensembles are performing better than their single counterpart.

8.2 Future Research Directions

In this thesis, we preferred the automatic annotation strategy over the manual annotation in order to generate a huge amount of labeled data. To improve the overall performance of all of our supervised systems, we think that it is necessary to be more accurate while generating the labeled data. If we can train our systems better, they will perform better while classifying the unseen data set. Therefore, we plan to work on finding more sophisticated approaches to effective automatic labeling so that we can experiment on different supervised methods. We will also evaluate our systems by providing manually annotated data during training.

We also plan to decompose the complex questions into several simple questions before measuring the similarity between the document sentence and the query sentence. This will certainly serve to create more limited trees and subsequences which might increase the

precision. Thus, we expect that by decomposing complex questions into the sets of sub-questions, the quality of answers returned by the system will improve and better coverage for the question as a whole will be achieved.

Integer Linear Programming (ILP) has recently attracted much attention in the NLP community. Most of these approaches use ILP to model problems in a more global manner. Capturing the global properties of a problem can improve the accuracy of a model as it is able to represent the long-range dependencies of the problem. So, we will apply ILP approaches in order to see how it works to answer complex questions.

References

Banko, M., V. Mittal, M. Kantrowitz, and J. Goldstein. 1999. Generating Extraction-based Summaries from Hand-Written Summaries by Aligning Text Spans. In *Proceedings of the Pacific Association for Computational Linguistics (PACLING 1999)*, Waterloo, Canada.

Barzilay, R. 2003. Sentence Alignment for Monolingual Comparable Corpora. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP 2003)*, pages 25–32, Sapporo, Japan.

Bennett, P. N., S. T. Dumais, and E. Horvitz. 2002. Probabilistic Combination of Text Classifiers Using Reliability Indicators: Models and Results. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002*, Tampere, Finland.

Berger, A. L., S. A. Della Pietra, and V. J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

Cancedda, N., E. Gaussier, C. Goutte, and J. M. Renders. 2003. Word Sequence Kernels. *Journal of Machine Learning Research*, 3:1059–1082.

Carbonell, J., Y. Geng, and J. Goldstein. 1997. Automated Query-Relevant Summarization and Diversity-based Reranking. In *International Joint Conference on Artificial Intelligence Workshop on AI in Digital Libraries (IJCAI 1997)*, pages 12–19, Japan.

Carbonell, J. and J. Goldstein. 1998. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 335–336, Melbourne, Australia.

Carbonell, J., D. Harman, E. Hovy, S. Maiorano, J. Prange, and K. Sparck-Jones. 2000. Vision Statement to Guide Research in Question & Answering (Q&A) and Text Summarization. *National Institute of Standards and Technology (NIST) Draft Publication*.

Chali, Y., S. A. Hasan, and S. R. Joty. 2009a. A SVM-Based Ensemble Approach to Multi-Document Summarization. In *Proceedings of the 22nd Canadian Conference on Artificial Intelligence (CAI 2009)*, pages 199–202, Kelowna, Canada. Springer-Verlag.

Chali, Y., S. A. Hasan, and S. R. Joty. 2009b. Do Automatic Annotation Techniques Have Any Impact on Supervised Complex Question Answering? In *Proceedings of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP 2009)*, pages 329–332, Suntec, Singapore.

Chali, Y., S. A. Hasan, and S. R. Joty. 2009c. Supervised Approaches to Complex Question Answering. In *Proceedings of the Pacific Association for Computational Linguistics (PACLING 2009)*, Sapporo, Japan.

Chali, Y. and S. R. Joty. 2007. Word Sense Disambiguation Using Lexical Cohesion. In *Proceedings of the 4th International Conference on Semantic Evaluations*, pages 476–479, Prague. ACL.

Chali, Y. and S. R. Joty. 2008. Selecting Sentences for Answering Complex Questions. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP 2008)*, pages 304–313, Hawaii, USA.

Chali, Y., S. R. Joty, and S. A. Hasan. 2009. Complex Question Answering: Unsupervised Learning Approaches and Experiments. *Journal of Artificial Intelligence Research*, 35:1–47.

Charniak, E. 1999. A Maximum-Entropy-Inspired Parser. In *Technical Report CS-99-12*, Brown University, Computer Science Department.

Collins, M. and N. Duffy. 2001. Convolution Kernels for Natural Language. In *Proceedings of Neural Information Processing Systems*, pages 625–632, Vancouver, Canada.

Conroy, J. M. and D. P. O'Leary. 2001. Text Summarization Via Hidden Markov Models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 406–407, New Orleans, LA, USA.

Cortes, C. and V. N. Vapnik. 1995. Support Vector Networks. *Machine Learning*, 20:273–297.

Das, D. and A. F. T. Martins. 2007. *A Survey on Automatic Text Summarization*. Language Technologies Institute, Carnegie Mellon University.

Deerwester, S., S. Dumais, G. Furnas, T. Landauer, and R. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Dietterich, T. G. 2000. Ensemble Methods in Machine Learning. pages 1–15. Springer-Verlag.

Dietterich, T. G. and G. Bakiri. 1995. Solving Multiclass Learning Problems Via Error Correcting Output Codes. *Journal of Artificial Intelligence Research*, 2:263–286.

Edmundson, H. P. 1969. New Methods in Automatic Extracting. *Journal of the Association for Computing Machinery (ACM)*, 16(2):264–285.

Efron, B. and R. J. Tibshirani. 1994. *An Introduction to the Bootstrap*. CRC Press.

Erkan, G. and D. R. Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Fellbaum, C. 1998. WordNet - An Electronic Lexical Database. Cambridge, MA. MIT Press.

Ferrier, L., 2001. *A Maximum Entropy Approach to Text Summarization*. M.Sc. thesis, School of Artificial Intelligence, Division of Informatics, University of Edinburgh.

Freund, Y. and R. E. Schapire. 1995. A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. In *European Conference on Computational Learning Theory*, pages 23–37.

Goldstein, J., M. Kantrowitz, V. Mittal, and J. Carbonell. 1999. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. In *Proceedings of the 22nd International ACM Conference on Research and Development in Information Retrieval (SIGIR 1999)*, pages 121–128, Berkeley, CA, USA.

Gong, Y. and X. Liu. 2001. Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis. In *Proceedings of the 24th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2001)*, pages 19–25, New Orleans, LA, USA.

Hacioglu, K., S. Pradhan, W. Ward, J. H. Martin, and D. Jurafsky. 2003. Shallow Semantic Parsing Using Support Vector Machines. In *Technical Report TR-CSLR-2003-03*, University of Colorado.

Harabagiu, S., F. Lacatusu, and A. Hickl. 2006. Answering Complex Questions with Random Walk Models. In *Proceedings of the 29th Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 220 – 227.

Hirao, T., H. Isozaki, E. Maeda, and Y. Matsumoto. 2002a. Extracting Important Sentences with Support Vector Machines. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 1–7, Taipei, Taiwan.

Hirao, T., Y. Sasaki, H. Isozaki, and E. Maeda. 2002b. NTT's Text Summarization System for DUC 2002. In *Proceedings of the Document Understanding Conference*, pages 104–107, Philadelphia, Pennsylvania, USA.

Hirao, T., J. Suzuki, H. Isozaki, and E. Maeda. 2003. NTT's Multiple Document Summarization System for DUC 2003. In *Proceedings of the Document Understanding Conference*, Edmonton, Canada.

Hirao, T., J. Suzuki, H. Isozaki, and E. Maeda. 2004. Dependency-based Sentence Alignment for Multiple Document Summarization. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 446–452, Geneva, Switzerland.

Hoi, C.H. and M.R. Lyu. 2004. Group-based Relevance Feedback with Support Vector Machine Ensembles. In *Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004)*, volume 3, pages 874–877.

Hovy, E., C. Y. Lin, and L. Zhou. 2005. A BE-based Multi-document Summarizer with Query Interpretation. In *Proceedings of the Document Understanding Conference*, Vancouver, B.C. Canada.

Hovy, E., C. Y. Lin, L. Zhou, and J. Fukumoto. 2006. Automated Summarization Evaluation with Basic Elements. In *Proceedings of the 5th Conference on Language Resources and Evaluation*, Genoa, Italy.

Hsu, C., C. Chang, and C. Lin. 2008. A Practical Guide to Support Vector Classification, National Taiwan University, Taipei 106, Taiwan, http://www.csie.ntu.edu.tw/ cjlin.

Jing, H. and K. R. McKeown. 1999. The Decomposition of Human-Written Summary Sentences. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*, pages 129–136, Berkeley, CA, USA. ACM.

Joachims, T. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *European Conference on Machine Learning (ECML)*.

Joachims, T. 1999. Making large-Scale SVM Learning Practical. In *Advances in Kernel Methods - Support Vector Learning*.

Jones, Karen S. and Julia Galliers. 1996. Evaluating Natural Language Processing Systems (An Analysis and Review). Secaucus, NJ, USA. Springer-Verlag New York, Inc.

Joty, S. R., 2008. *Answer Extraction for Simple and Complex Questions*. M.Sc. Thesis, Department of Computer Science, University of Lethbridge, Canada.

Jurafsky, D. and J. H. Martin, 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.

Kingsbury, P. and M. Palmer. 2002. From Treebank to PropBank. In *Proceedings of the International Conference on Language Resources and Evaluation*, Las Palmas, Spain.

Kolla, M., 2004. *Automatic Text Summarization using Lexical Chains: Algorithms and Experiments*. M.Sc. Thesis, Department of Computer Science, University of Lethbridge, Canada.

Kudo, T. and Y. Matsumoto. 2001. Chunking with Support Vector Machine. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2001)*, pages 192–199, Carnegie Mellon University, Pittsburgh, PA, USA.

Kupiec, J. 1992. Robust Part-Of-Speech Tagging Using a Hidden Markov Model. *Computer Speech and Language*, 6(3):225–242.

Kupiec, J., J. Pedersen, and F. Chen. 1995. A Trainable Document Summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1995)*, pages 68–73, Seattle, Washington, USA.

Lafferty, J., A. K. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning (ICML 2001)*, Williamstown, MA, USA.

Li, J., L. Sun, C. Kit, and J. Webster. 2007. A Query-Focused Multi-Document Summarizer Based on Lexical Chains. In *Proceedings of the Document Understanding Conference*, Rochester, USA. NIST.

Lin, C. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of Association for Computational Linguistics*, pages 74–81, Barcelona, Spain.

Lodhi, H., C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. 2002. Text Classification Using String Kernels. *Journal of Machine Learning Research*, 2:419–444.

Luhn, H. P. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2:159–165.

Mani, I., 2001. *Automatic Summarization*. John Benjamins Co, Amsterdam/Philadelphia.

Mani, I. and M. T. Maybury, 1999. *Advances in Automatic Text Summarization*. MIT Press.

Marcu, D. 1999. The Automatic Construction of Large-scale Corpora for Summarization Research. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*, pages 137–144, Berkeley, CA, USA.

McCallum, A. K. 2002. MALLET: A Machine Learning for Language Toolkit.

Mihalcea, R. and P. Tarau. 2004. TextRank: Bringing Order into Texts. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP 2004)*, Barcelona, Spain.

Molla, D. and S. Wan. 2006. Macquarie University at DUC 2006: Question Answering for Summarisation. In *Proceedings of the Document Understanding Conference*. NIST.

Moschitti, A. and R. Basili. 2006. A Tree Kernel Approach to Question and Answer Classification in Question Answering Systems. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, Genoa, Italy.

Moschitti, A., S. Quarteroni, R. Basili, and S. Manandhar. 2007. Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classificaion. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pages 776–783, Prague, Czech Republic.

Nguyen, M. L., A. Shimazu, X. H. Phan, T. B. Ho, and S. Horiguchi. 2005. Sentence Extraction with Support Vector Machine Ensemble. In *First World Congress of the International Federation for Systems Research (IFSR'05), Symposium on Data/Text Mining from Large Databases*, Kobe.

Nigam, K., J. Lafferty, and A. McCallum. 1999. Using Maximum Entropy for Text Classication. In *International Joint Conference on Artificial Intelligence (IJCAI-99) Workshop on Machine Learning for Information Filtering*, Stockholm, Sweden.

Orasan, C. 2005. Automatic Annotation of Corpora for Text Summarization: A Comparative Study. In *Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2005)*, pages 670–681, Mexico City, Mexico.

Otterbacher, J., G. Erkan, and D. R. Radev. 2005. Using Random Walks for Question-focused Sentence Retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 915–922, Vancouver, Canada.

Parmanto, B., P. W. Munro, and H. R. Doyle. 1996. Improving Committee Diagnosis with Resampling Techniques. In *Advances in Neural Information Processing Systems*, volume 8, pages 882–888.

Pasca, M. and S. M. Harabagiu. 2001. Answer Mining from On-Line Documents. In *Proceedings of the Association for Computational Linguistics 39th Annual Meeting and 10th Conference of the European Chapter Workshop on Open-Domain Question Answering*, pages 38–45, Toulouse, France.

Pla, F., A. Molina, and N. Prieto. 2000. Improving Text Chunking by Means of Lexical-Contextual Information in Statistical Language Models. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL 2000*, Lisbon, Portugal.

Qi, H. and M. Huang. 2007. Research on SVM Ensemble and Its Application to Remote Sensing Classification. In *Proceedings of the International Conference on Intelligent Systems and Knowledge Engineering (ISKE 2007)*, Chengdu, China.

Rabiner, L. and B. Juang. 1993. Fundamentals of Speech Recognition. *Prentice Hall Signal Processing Series,Prentice Hall,Inc.*

Rabiner, L. R. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)*, pages 257–285.

Rooney, N., D. W. Patterson, S. S. Anand, and A. Tsymbal. 2004. Random Subspacing for Regression Ensembles. In *Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*, Miami Beach, Florida.

Sekine, S. 2002. Proteus Project OAK System (English Sentence Analyzer), http://nlp.nyu.edu/oak.

Sekine, S. and C. A. Nobata. 2001. Sentence Extraction with Information Extraction Technique. In *Proceedings of the Document Understanding Conference (DUC 2001)*, New Orleans, Louisiana, USA.

Shen, D., J. Sun, H. Li, Q. Yang, and Z. Chen. 2007. Document Summarization Using Conditional Random Fields. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 2862–2867, Hyderabad, India.

Silva, C. and B. Ribeiro. 2006. Rare Class Text Categorization with SVM Ensemble. *Journal of Electrotechnical Review (Przeglad Elektrotechniczny)*, 1:28–31.

Strzalkowski, T. and S. Harabagiu, 2008. *Advances in Open Domain Question Answering*. Springer.

Todorovski, L. and S. Dzeroski. 2000. Combining Multiple Models with Meta Decision Trees. In *Proceedings of Principles of Data Mining and Knowledge Discovery*, pages 54–64.

Toutanova, K., C. Brockett, M. Gamon, J. Jagarlamudi, H. Suzuki, and L. Vanderwende. 2007. The pythy summarization system: Microsoft research at duc 2007. In *Proceedings of the Document Understanding Conference (DUC 2007)*, Rochester, USA. NIST.

Toutanova, K., F. Chen, K. Popat, and T. Hofmann. 2001. Text Classification in a Hierarchical Mixture Model for Small Training Sets. In *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM 2001)*, Atlanta, Georgia.

Vapnik, V. N. 1998. *Statistical Learning Theory*. Wiley Interscience.

Wallach, H., 2002. *Efficient Training of Conditional Random Fields*. M.Sc. thesis, Division of Informatics, University of Edinburgh.

Wei, L. and W.X. Zhang. 2004. Classification Based on SVM Ensemble. In *Computer Engineering*, volume 30, pages 1–2.

Wolpert, D. H. 1992. Stacked Generalization. *Neural Networks*, 5:241–259.

Wong, K., M. Wu, and W. Li. 2008. Extractive Summarization Using Supervised and Semi-supervised Learning. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 985–992, Manchester.

Yan, R., A. Hauptmann, R. Jin, and Y. Liu. 2003. On Predicting Rare Class with SVM Ensemble in Scene Classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2003)*.

Zhang, A. and W. Lee. 2003. Question Classification Using Support Vector Machines. In *Proceedings of the Special Interest Group on Information Retrieval*, pages 26–32, Toronto, Canada. ACM.

Appendix-A
Reference Summaries

Topic

<topic>
<num> D0701A </num>
<title> Southern Poverty Law Center </title>
<narr>
Describe the activities of Morris Dees and the Southern Poverty Law Center.
</narr>
</topic>

Human Produced Summaries

1.

Morris Dees was co-founder of the Southern Poverty Law Center (SPLC) in 1971 and has served as its Chief Trial Counsel and Executive Director. The SPLC participates in tracking down hate groups and publicizing their activities in its Intelligence Report, teaching tolerance and bringing lawsuits against discriminatory practices and hate groups. As early as 1973 the SPLC won a federal case which forced funeral homes throughout the U.S. to provide equal services to blacks and whites. In 1991 it started a classroom program "Teaching Tolerance" which features books, videos, posters and a magazine that goes to more than 400,000 teachers. It also funded a civil rights litigation program in Georgia to provide free legal assistance to poor people. The SPLC's most outstanding successes, however, have been in its civil lawsuits against hate groups. Dees and the SPLC have fought to break the organizations by legal action resulting in severe financial penalties. Described as "wielding the civil lawsuit like a Buck Knife, carving financial assets out of hate group leaders," the technique has been most impressive: 1987-$7 million against the United Klans of America in Mobile, Alabama; 1989-$1 million against Klan groups in Forsyth County, Georgia; 1990-$9 million against the White Aryan Resistance in Portland, Oregon; and 1998-$20 million against The Christian Knights of the Ku Klux Klan in Charleston, South Carolina. But despite these judgments the Ku Klux Klan and White Aryan Resistance have survived.

2.

Morris Dees is a co-founder and leader of the Southern Poverty Law Center, located in Montgomery, Alabama. It was founded to battle racial bias and has expanded its efforts by tracking hate crimes and the increasing spread of racist organizations across the US. "Teaching Tolerance" is a major program of the Center. Under that program, a magazine promoting interracial and intercultural understanding goes to more than 400,000 teachers. Other publications of the Center include the magazine "Intelligence Report" and pamphlets "Ten Ways to Fight Hate" and "Fighting Hate at School". Dees has determined that the civil courts

are an effective forum in which to attack and destroy hate groups. He has used the civil lawsuit like a "Buck Knife, carving financial assets out of hate group leaders". Some skeptics thought that Dees sought out victims of hate groups to profit from their tragedy. However, Dees does not charge the groups and the Center estimates that it collects only 2% on successful judgments. Dees has a perfect record in the major lawsuits he has prosecuted. Successful judgments include one for $21.5M against a South Carolina branch of the Ku Klux Klan for burning the Macedonia Baptist Church. Others include $6.3M against Aryan Nation's leader Richard Butler and $7M against a Klan group that killed a black man in Mobile, Alabama. The Center operates mostly on contributions that in the late 1990s have increased to around $100 Million annually.

3.

The Southern Poverty Law Center, a non-profit organization in Montgomery, Alabama, was founded in the 1970s to help minorities litigate against civil rights abuses. Located in the same block as Dexter Avenue Baptist Church, which was once pastored by the Reverend Martin Luther King Jr., the center has effectively established programs and implemented actions over the last three decades towards fulfilling its mission. A core initiative for the center is a classroom program started in 1991 called "Teaching Tolerance," that involves more than 400,000 teachers and includes materials promoting "interracial and intercultural understanding". The SPLC also funds a three-year, $100,000 civil rights litigation program in Georgia designed to stem federal cutbacks in programs that provide free legal assistance to poor people in civil actions. Additionally, the center produces the Intelligence Report, a magazine that tracks hate groups and covers right-wing extremists. The law center's co-founder and chief trial counsel, Morris Dees, has successfully handled civil rights cases for more than 30 years-- in 1973 his federal lawsuit had the practical effect of forcing funeral homes to provide equal services to blacks and whites. As a white lawyer, Dees has been instrumental in crusading against racial intolerance by using lawsuits to destroy the finances of hate groups. Since 1979 he has won a series of six, countrywide civil rights suits against the Ku Klux Klan and other Neo-Nazi groups accused of criminal activity. In every case,Dees secured multi-million dollar judgments against the convicted defendants to effectively put them out of business.

4.

The Southern Poverty Law Center is a nonprofit research group based in Montgomery, Alabama that battles racial bias. It tracks US hate crimes and the spread of racist organizations. It covers right-wing

extremists in its magazine Intelligence Report. Through its Teaching Tolerance program it provides materials to teachers to promote interracial and intercultural understanding. It freely distributes booklets on combating hate to schools, mayors, police chiefs, and other interested groups and citizens. It advises city leaders faced with hate crimes. Morris Dees co-founded the SPLC in 1971 and is its chief trial counsel and executive director, following Julian Bond. Dees and the SPLC seek to destroy hate groups through multi-million dollar civil suits that go after assets of groups and their leaders. In six lawsuits based on hate crimes or civil rights abuses, they have never lost. They successfully sued the Ku Klux Klan and the related Invisible Empire Klan, United Klan of America, and Christian Knights of the KKK; the White Aryan Resistance; and the Aryan Nations and its founder Richard Butler. The SPLC influenced funeral homes to provide equal services to blacks and whites, tried to discourage the sale and distribution of the racist book The Turner Diaries, and protected Vietnamese fishermen from Klan intimidation. The SPLC devotes much effort to raising the funds needed to help minorities litigate against civil rights abuses. It charges its clients nothing. Nearly all money from settlements goes to the victims, with less than 2 percent going to the SPLC.

HMM Generated Summary (ESSK labeled)

Morris Dees , the co-founder of the Southern Poverty Law Center in Montgomery , Ala. , and one of the attorneys for the plaintiffs , said he intended to enforce the judgment , taking everything the Aryan Nations owns , including its trademark name . The Southern Poverty Law Center , which was founded in the 1970s to battle racial bias , won major legal fights against the Ku Klux Klan and other white supremacist groups . Lawyer Morris Dees , the co-founder of the Southern Poverty Law Center who is representing Victoria Keenan and Victoria Keenan 's son , Jason , introduced letters , photographs and depositions to contradict the men 's testimony . The notice was the first indication that the lawsuit , brought by the Southern Poverty Law Center , may drive the group out of Idaho. '' I have been asked if I would continue to host the yearly National Congress and my answer was , of course , an astounding YES ! '' wrote August B. Kreis III , Web master for the Aryan Nations and a Posse Comitatus leader in Pennsylvania . In his suit here , Dees , a founder of the Southern Poverty Law Center , seeks unspecified damages on behalf of a woman and her son , both white , who were attacked by guards near the compound in July 1998 . I directed him to Julian Bond , who was then president of the Southern Poverty Law Center . But the book did n't begin , nor will it end with the King trial , as a report by the Montgomery , Ala.-based Southern Poverty Law Center demonstrates.

MaxEnt Generated Summary (ESSK labeled)

The notice was the first indication that the lawsuit , brought by the Southern Poverty Law Center , may drive the group out of Idaho. '' I have been asked if I would continue to host the yearly National Congress and my answer was , of course , an astounding YES ! '' wrote August B. Kreis III , Web master for the Aryan Nations and a Posse Comitatus leader in Pennsylvania . Morris Dees , the co-founder of the Southern Poverty Law Center in Montgomery , Ala. , and one of the attorneys for the plaintiffs , said he intended to enforce the judgment , taking everything the Aryan Nations owns , including its trademark name . Randall Lee Rojas identified the reporter as Trish O'Kane , who writes for Intelligence Report , a magazine of the Southern Poverty Law Center , a civil rights group in Montgomery , Ala. Morris Dees , co-founder of the Southern Poverty Law Center in Montgomery , Ala. , represented the Keenans and has said Keenans intends to take everything the Aryan Nations owns to pay the judgment , including the sect 's name . Since co-founding the Southern Poverty Law Center in 1971 , Dees has wielded the civil lawsuit like a buck knife , carving financial assets out of hate group leaders who inspire followers to beat , burn and kill . The Southern Poverty Law Center represented the predominantly black Macedonia Baptist Church in Clarendon County , which won a $ 21.5 million judgment against the Christian Knights and Klan members after the church was burned four years ago .

CRF Generated Summary (ESSK labeled)

He is convinced the best weapons to combat organized hate are a finely honed legal complaint and a fully loaded bank account . Butler moved to northern Idaho from California in 1973 , and began holding an annual event called the Aryan Nations Congress in 1981 . O'Connor actions were so jaw-droppingly over , it was hard not to admire their raw loopiness , though they overshadowed O'Connor 's talent and undermined record sales . Lawyers for two of Rojas ' co-defendants , Ritch Bryant and Jessica Colwell , sought the material as evidence that their clients had no hand in the Nov. 25 , 1995 , beating death of a black man , Milton Walker Jr. A `` who 's who of the hate movement '' visited the place over the years , including William Pierce , author of a book about a race war called The Turner Diaries that allegedly inspired Oklahoma City bomber Timothy McVeigh ; former Texas Klan leader Louis Beam ; and Tom Metzger of the White Aryan Resistance , who lost his house and possessions to a SPLC lawsuit . In Birmingham , at a recent ceremony at the 16th Street Church commemorating the victims of the bombing , police officers stood guard outside . But the civil rights organization still spends most of its money

on programs to help poor people . `` I respect myself for being the age that I was and having this passion and integrity that I did have , '' she says . But the group persisted . Quickly , he changes the subject . In short , chicks rule .

SVM Generated Summary (ESSK labeled)

But the book did n't begin , nor will it end with the King trial , as a report by the Montgomery , Ala.-based Southern Poverty Law Center demonstrates . He said Gilliam also told the informant someone should kill the FBI sniper who killed the wife of white supremacist Randy weaver during an 11-day standoff in 1992 at Ruby Ridge , Idaho , along with civil rights lawyer Morris Dees of the Montgomery-based Southern Poverty Law Center . Randall Lee Rojas identified the reporter as Trish O'Kane , who writes for Intelligence Report , a magazine of the Southern Poverty Law Center , a civil rights group in Montgomery , Ala. Morris Dees , the chief trial counsel of the Southern Poverty Law Center and a member of the original team of lawyers that handled the case , said , `` Although the case was not binding , because it never reached the Supreme Court , it served notice to funeral homes , and even cemeteries and other businesses , that if they practiced discrimination against blacks they could be violating federal law . '' But the larger purpose of the lawsuit was to bankrupt the Aryan Nations compound , limit Butler 's ability to spread a gospel of racial hate and persuade the jury to `` return a verdict that will be heard all over this nation , '' a lawyer for the Keenans , Dees said in closing arguments . Six distributors of skinhead music are donating proceeds from the sale of CDs with titles like Morris Dees for You , '' and '' Holocaust 2000 . ''

Ensemble Generated Summaries (ROUGE labeled)

1. Heterogeneous

Morris Dees , the chief trial counsel of the Southern Poverty Law Center and a member of the original team of lawyers that handled the case , said , `` Although the case was not binding , because it never reached the Supreme Court , it served notice to funeral homes , and even cemeteries and other businesses , that if they practiced discrimination against blacks they could be violating federal law . '' Lawyers from the Southern Poverty Law Center , a civil rights organization in Montgomery , Ala. , advanced his case in federal court , charging that the Escude Funeral Home and Hixson Brothers Funeral Home in Avoyelles Parish either refused to deal with blacks or offered `` distinctly inferior services '' for the same prices that they charged whites . The SPLC , headed by Morris Dees in Alabama , is known for having

used civil law to break the back of the Ku Klux Klan . Morris Dees , the civil rights lawyer who led the plaintiffs ' legal team , has said he expected the judgment to bring a quick end to the Aryan Nations and its racist , anti-Semitic message . As a law student in 1977 Ku Klux Klan skipped classes for a week to watch Bill Baxley , who was the Alabama attorney general , successfully prosecute the Klan leader Robert Chambliss for murder in the bombing . Butler 's lawyer , Edgar Steele , argued throughout the six-day trial that however offensive jurors might find the views of Butler , Richard Girnt Butler should not be held responsible for the actions of a group of drunken young men .

2. Homogeneous

Morris Dees , the civil rights lawyer who led the plaintiffs ' legal team , has said he expected the judgment to bring a quick end to the Aryan Nations and its racist , anti-Semitic message . That is no more reasonable than trying to distinguish the ` good ' Jews from the bad ones _ or , as some of our thicker-skulled ` good ol' boys ' still insist on trying , separating the ` good niggers ' from the rest of their race . '' Morris Dees , the chief trial counsel of the Law Center , said he is surprised by what appears to be the increasing frequency and viciousness of such attacks . Crime Continues to Decline Violent crime in the United States dropped last year to its lowest level since the government began its annual national crime survey 26 years ago . From his compound , which is valued at about $ 200,000 and has a sign out front that reads Morris Dees only , '' Butler mails his literature , recruits followers and plays host to the annual Aryan World Congress , a skinhead symposium that often draws more than 100 acolytes . Clinton said more time is also needed to find a diplomatic solution to what has been a growing confrontation between the United States and Russia and China , staunch opponents to a U.S. missile defense . So in the current issue of The Source , O'Connor pumps `` Order in the Court , '' O'Connor 's first CD in five years , and gives a shout-out to the late Biggie Smalls .

Appendix-B
Reference Cue Words and Stop Words

Cue Words

indeed	further	as well
as this	either	neither
not only	but also	the reason is
as well as	also	moreover
what is more	as a matter of fact	furthermore
in addition	besides	to tell you the truth
in fact	actually	amazingly
to say nothing of	too	let alone
much less	additionally	nor
alternatively	on the other hand	not to mention
such as	this time	at this time
this also	several years ago	long ago
during	eventually	meanwhile
essentially	enormously	majority of the
absolutely	necessary	especially
specially	after	before
at least	at most	most
therefore	this is	that is
reasonable	according to	throughout
at this point	along with	previously
as	particularly	including
as an illustration	for example	like
in particular	for one thing	to illustrate
for instance	notably	by way of example
speaking about	considering	regarding
with regards to	as for	concerning
on the subject of	the fact that	similarly
in the same way	by the same token	in a like manner
equally	likewise	namely
specifically	thus	I mean
put another way	in other words	but
by way of contrast	while	on the other hand
however	yet	whereas
though	in contrast	when in fact
conversely	still	even more
above all	more importantly	but even so
nevertheless	even though	admittedly
nonetheless	despite	notwithstanding

albeit	although	in spite of
regardless	granted	be that as it may
either way	whichever happens	in either event
in any case	at any rate	in either case
whatever happens	all the same	in any event
rather	instead	for the reason that
being that	in view of	inasmuch as
because	seeing that	owing to
due to	in that	since
forasmuch as	for this reason	on condition
provided that	in case	in the event that
as long as	so long as	unless
given that	granting	providing that
even if	only if	as a result
consequently	hence	in consequence
so that	accordingly	as a consequence
so much that	for the purpose of	in the hope that
for fear that	with this intention	to the end that
in order to	lest	with this in mind
in order that	so as to	under those circumstances
then	in that case	if not
that being the case	if so	otherwise
initially	to start with	first of all
to begin with	at first	for a start
secondly	subsequently	previously
next	afterwards	to conclude
as a final point	at last	in the end
finally	lastly	to change the topic
incidentally	by the way	to get back to the point
to resume	anyhow	anyway
at any rate	to return to the subject	as was previously stated
in summary	all in all	to make a long story short
as I have said	to sum up	overall
as has been mentioned	to summarize	to be brief
briefly	given these points	in all
on the whole	as has been noted	hence
in conclusion	in a word	to put it briefly
in sum	altogether	in short

Stop Words

reuters	ap	jan	feb	mar	apr
may	jun	jul	aug	sep	oct
nov	dec	tech	news	index	mon
tue	wed	thu	fri	sat	’s
a	a’s	able	about	above	according
accordingly	across	actually	after	afterwards	again
against	ain’t	all	allow	allows	almost
alone	along	already	also	although	always
am	amid	among	amongst	an	and
another	any	anybody	anyhow	anyone	anything
anyway	anyways	anywhere	apart	appear	appreciate
appropriate	are	aren’t	around	as	aside
ask	asking	associated	at	available	away
awfully	b	be	became	because	become
becomes	becoming	been	before	beforehand	behind
being	believe	below	beside	besides	best
better	between	beyond	both	brief	but
by	c	c’mon	c’s	came	can
can’t	cannot	cant	cause	causes	certain
certainly	changes	clearly	co	com	come
comes	concerning	consequently	consider	considering	contain
containing	contains	corresponding	could	couldn’t	course
currently	d	definitely	described	despite	did
didn’t	different	do	does	doesn’t	doing
don’t	done	down	downwards	during	e
each	edu	eg	e.g.	eight	either
else	elsewhere	enough	entirely	especially	et
etc	etc.	even	ever	every	everybody
everyone	everything	everywhere	ex	exactly	example
except	f	far	few	fifth	five
followed	following	follows	for	former	formerly
forth	four	from	further	furthermore	g
get	gets	getting	given	gives	go
goes	going	gone	got	gotten	greetings
h	had	hadn’t	happens	hardly	has
hasn’t	have	haven’t	having	he	he’s
hello	help	hence	her	here	here’s
hereafter	hereby	herein	hereupon	hers	herself
hi	him	himself	his	hither	hopefully

how	howbeit	however	i	i'd	i'll
i'm	i've	ie	i.e.	if	ignored
immediate	in	inasmuch	inc	indeed	indicate
indicated	indicates	inner	insofar	instead	into
inward	is	isn't	it	it'd	it'll
it's	its	itself	j	just	k
keep	keeps	kept	know	knows	known
l	lately	later	latter	latterly	least
less	lest	let	let's	like	liked
likely	little	look	looking	looks	ltd
m	mainly	many	may	maybe	me
mean	meanwhile	merely	might	more	moreover
most	mostly	mr.	ms.	much	must
my	myself	n	namely	nd	near
nearly	necessary	need	needs	neither	never
nevertheless	new	next	nine	no	nobody
non	none	noone	nor	normally	not
nothing	novel	now	nowhere	o	obviously
of	off	often	oh	ok	okay
old	on	once	one	ones	only
onto	or	other	others	otherwise	ought
our	ours	ourselves	out	outside	over
overall	own	p	particular	particularly	per
perhaps	placed	please	plus	possible	presumably
probably	provides	q	que	quite	qv
r	rather	rd	re	really	reasonably
regarding	regardless	regards	relatively	respectively	right
s	said	same	saw	say	saying
says	second	secondly	see	seeing	seem
seemed	seeming	seems	seen	self	selves
sensible	sent	serious	seriously	seven	several
shall	she	should	shouldn't	since	six
so	some	somebody	somehow	someone	something

sometime	sometimes	somewhat	somewhere	soon	sorry
specified	specify	specifying	still	sub	such
sup	sure	t	t's	take	taken
tell	tends	th	than	thank	thanks
thanx	that	that's	thats	the	their
theirs	them	themselves	then	thence	there
there's	thereafter	thereby	therefore	therein	theres
thereupon	these	they	they'd	they'll	they're
they've	think	third	this	thorough	thoroughly
those	though	three	through	throughout	thru
thus	to	together	too	took	toward
towards	tried	tries	truly	try	trying
twice	two	u	un	under	unfortunately
unless	unlikely	until	unto	up	upon
us	use	used	useful	uses	using
usually	uucp	v	value	various	very
via	viz	vs	w	want	wants
was	wasn't	way	we	we'd	we'll
we're	we've	welcome	well	went	were
weren't	what	what's	whatever	when	whence
whenever	where	where's	whereafter	whereas	whereby
wherein	whereupon	wherever	whether	which	while
whither	who	who's	whoever	whole	whom
whose	why	will	willing	wish	with
within	without	won't	wonder	would	would
wouldn't	x	y	yes	yet	you
you'd	you'll	you're	you've	your	yours
yourself	yourselves	z	zero		