ABSTRACT

Title of dissertation: MULTIPLE ALTERNATIVE SENTENCE COMPRESSIONS AS A TOOL FOR AUTOMATIC SUMMARIZATION TASKS

David M. Zajic
Doctor of Philosophy, 2007

Dissertation directed by: Professor Bonnie J. Dorr, advisor
Professor Jimmy Lin, co-advisor
Department of Computer Science

Automatic summarization is the distillation of important information from a source into an abridged form for a particular user or task. Many current systems summarize texts by selecting sentences with important content. The limitation of extraction at the sentence level is that highly relevant sentences may also contain non-relevant and redundant content.

This thesis presents a novel framework for text summarization that addresses the limitations of sentence-level extraction. Under this framework text summarization is performed by generating Multiple Alternative Sentence Compressions (MASC) as candidate summary components and using weighted features of the candidates to construct summaries from them. Sentence compression is the rewriting of a sentence in a shorter form. This framework provides an environment in which hypotheses about summarization techniques can be tested.

Three approaches to sentence compression were developed under this framework. The first approach, HMM Hedge, uses the Noisy Channel Model to calculate the most likely compressions of a sentence. The second approach, Trimmer, uses syntactic trimming rules that are linguistically motivated by Headlinese, a form of compressed English associated with newspaper headlines. The third approach, Topiary, is a combination of fluent text with topic terms.

The MASC framework for automatic text summarization has been applied to the tasks of headline generation and multi-document summarization, and has been used for initial work in summarization of novel genres and applications, including broadcast news, email threads, cross-language, and structured queries. The framework supports combinations of component techniques, fostering collaboration between development teams.

Three results will be demonstrated under the MASC framework. The first is that an extractive summarization system can produce better summaries by automatically selecting from a pool of compressed sentence candidates than by automatically selecting from unaltered source sentences. The second result is that sentence selectors can construct better summaries from pools of compressed candidates when they make use of larger candidate feature sets. The third result is that for the task of Headline Generation, a combination of topic terms and compressed sentences performs better then either approach alone. Experimental evidence supports all three results.

MULTIPLE ALTERNATIVE SENTENCE COMPRESSIONS AS A TOOL FOR AUTOMATIC SUMMARIZATION TASKS

by

David M. Zajic

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2007

Advisory Committee:
Professor Bonnie J. Dorr, Chair/Advisor
Professor Jimmy Lin, Co-Chair/Co-Advisor
Professor William Gasarch
Professor Dianne O'Leary
Professor Lawrence Moss
Richard M. Schwartz

Acknowledgments

My first thanks must go to my advisors Bonnie Dorr, Jimmy Lin, and Rich Schwartz for their encouragement, guidance, research wisdom, and, of course, advice on the broadest range of topics. I also thank my committee members: Dianne O'Leary, Bill Gasarch, and Larry Moss for their careful reading of my dissertation, constructive feedback, and insightful questions. Philip Resnik and Doug Oard have also been consistently excellent role models.

My graduate career has been enriched by the friendship, technical advice, and faith of my fellow students Nizar Habash, Mona Diab, Okan Kolak, Matt Snover, Stacy Hobson, Nate Waisbrot, and Nitin Madnani: you have broadened my world. I thank my colleagues from MITRE Corporation, Flo Reeder and Keith Miller, who showed it could be done.

My sincere gratitude to Sean Boisen and the BBN Columbia office for graciously providing logistical support and a friendly place to work.

I have had the good fortune to collaborate with excellent researchers from other institutions: Judith Schlesinger and John Conroy of IDA/CCS, Radu Florian of IBM, and Jon Elsas of CMU.

My gratitude goes to Erin Greenwell, Paul Li, Steve Martin, Phyllis Florian, Hunter Provyn, Matt Snover, Nate Waisbrot, Nizar Habash, Nitin Madnani, Naomi Chang, Ken Zajic, Lois Zajic, Jon Teske, Joyce Teske, and everyone else who annotated data, wrote summaries, made relevance judgments, and otherwise provided data for me to analyze.

Finally, I thank my parents, Ken and Lois Zajic, and my fiancée, Naomi Chang, for their patience, love and support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Automatic summarization is the process of creating a short document that can serve as a surrogate of a longer document to meet a user's information need. The earliest efforts at automatic summarization consisted of extracting sentences from a source document based on the distribution of words. Sentences are selected for inclusion in a summary on the basis of the importance of the information they contain. Information is considered important if, for example, it is highly central to the document, or if it is highly relevant to user's information need. In the case of multi-sentence summaries, sentences should be selected to avoid redundancy of information.

The limitation of a sentence-extraction approach is that the granularity of the extracts is at the sentence level. A highly central or relevant sentence may additionally contain non-central or non-relevant information, or a mixture of novel and redundant information with respect to other sentences in the summary. Summaries are required in situations where space is constrained, so non-relevant and redundant information should not be allowed to take up space in the summary.

Sentence compression is a powerful tool for creating summaries that do not waste space. It is the rewriting of sentences from the source document so that they take up fewer characters. This is a difficult task because compression should preserve important information, readability, coherence and correctness of a sentence while only removing syntactically optional and semantically unimportant or redundant elements. The problem with this goal, as with summarization in general, is the notion of a truly generic summary.

Consider the following news story from the Washington Post, June 10, 2003:

> A D.C. police officer and a man with a rifle exchanged gunfire on a Northeast Washington street in a bizarre confrontation yesterday afternoon that ended with the gunman stripping himself naked and a police dog biting another officer. . . .
> The only injuries were scratches suffered by the alleged gunman, Damien J. Lee, 26, and a minor bite to the officer's knee. . . .
> The man was blocking traffic, she said, and pointing the rifle – described by police as an M1 carbine with the stock sawed off – at nearby people and cars. He was yelling profanities, Williams said, and saying, "I'm God."

Is this a story about police dogs, traffic disruptions or people who claim to be God? Researchers interested in police dogs, gun violence, mental illness, and commuter traffic issues will require different summaries of this document to make a decision whether this incident is one they want to learn more about. The copy editor at the Post decided to provide this article with two headlines:

> Gunman Says, 'I'm God,' Blocks Traffic, Fires, Strips
> Police Dog Bites Officer on the Knee In Struggle to Arrest Naked NE Man

Now consider how the first sentence of this example one could be compressed to fit in a 50 character window.[1] Here are three distinct versions.

(1) (i) Police officer and man with rifle exchange gunfire
(ii) Confrontation ended with gunman stripping himself
(iii) Confrontation ended with police dog biting officer

[1] For example, the screen of a personal digital assistant.

This example illustrates that both at the level of the document and the sentence, a piece of text may be relevant to many diverse interests, and that a single all-purpose summary cannot serve the needs of every reader.

There are several automatic summarization systems that make use of sentence compression: (Jing, 2000; Daumé and Marcu, 2005; Knight and Marcu, 2002; Banko et al., 2000; Turner and Charniak, 2005; Conroy et al., 2006b; Melli et al., 2006; Vanderwende et al., 2006). With the exception of Vanderwende et al. (2006), all of these systems produce only a single compressed version for a sentence. This dissertation proposes Multiple Alternative Sentence Compressions (MASC), or the generation of multiple candidate compressions for a sentence, as a novel contribution to the field of automatic summarization.

A problem that can occur with sentence extraction and sentence compression is loss of context. Consider the following sentence extracted from a story in the New York Times, November 8, 2006. Sentence (2i) is a manual compression of the source sentence to fewer than 50 characters.

(2) Though Bush affectionately patted Rumsfeld on the shoulder as he ushered him out of the Oval Office, there was little sugarcoating the reality that the defense chief, 74, was being offered as a sacrificial lamb amid the repudiation of Bush and his Iraq policy that the American electorate delivered on Tuesday.

(i) defense chief offered as sacrificial lamb

(ii) RUMSFELD defense chief offered as sacrificial lamb

The compression does retain the main event of the sentence, but because Rumsfeld's name has been removed, a reader without any other context would not know which defense chief had been let go. Fluent text is good at telling what happened, but often fails to give adequate context. The addition of the topic term

RUMSFELD in Sentence (2ii) provides a context that complements the fluent text. The combination of topic terms and fluent text is implemented in the Topiary system, and has been recognized in the community as a state-of-the-art system for very short summarization, or headline generation. Topiary will be presented in Section 4.3.

Note that the compression in Sentence (2i) is not fully grammatical in standard English. However it is fully comprehensible, because it follows the rules of Headlinese, a form of compressed English associated with newspaper headlines. A novel feature of the work presented here is the deliberate emulation of Headlinese in sentence compression.

1.1 Multiple Alternative Sentence Compression (MASC) Framework

This dissertation presents five summarization systems. Three perform the single-document summarization task, and two perform multi-document summarization. All five share a common underlying framework, Multiple Alternative Sentence Compressions (MASC), represented in Figure 1.1. The MASC framework divides the summarization task into three main stages: filtering, compression, and selection.

Filtering is the selection of text segments from the source document or documents for submission to the sentence compression stage. Usually the documents are pre-processed with a sentence boundary detector, and the text segments are sentences. However some sentence compression tools are amenable to contiguous blocks of words regardless of sentence boundary. Filtering could trivially consist

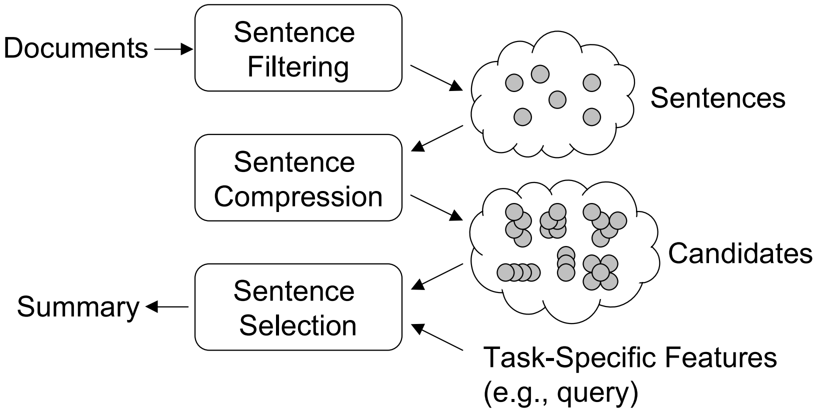


Figure 1.1: The Multiple Alternative Sentence Compression (MASC) framework for automatic summarization

of submitting all the sentences to the compression stage. However, aside from the processing time issues involved, a failure to use a filtering stage can actually harm performance. In the domain of written news, there is a such a strong bias for putting important information near the front of a document that the best approach so far has been to filter through only the first N sentences of a document, with N typically set to 1.

The use of sentence compression as a tool for automatic summarization is one of the central contributions of this dissertation. The sentence compression systems presented share the underlying approach of selecting words in order from a sentence (or block of words) with morphological variation. Three sentence compression tools using this approach have been implemented: HMM Hedge, Trimmer, and Topiary.

HMM Hedge uses separate language models of newspaper headlines and newspaper stories and a dynamic programming algorithm to find the subset of words from a block of text that is the most probable compressed version of a source sentence. The architecture of HMM Hedge is shown in Figure 1.2.[2] HMM Hedge can oper-

[2] In Figures 1.2 through 1.5, modules with solid lines represent software developed by the author

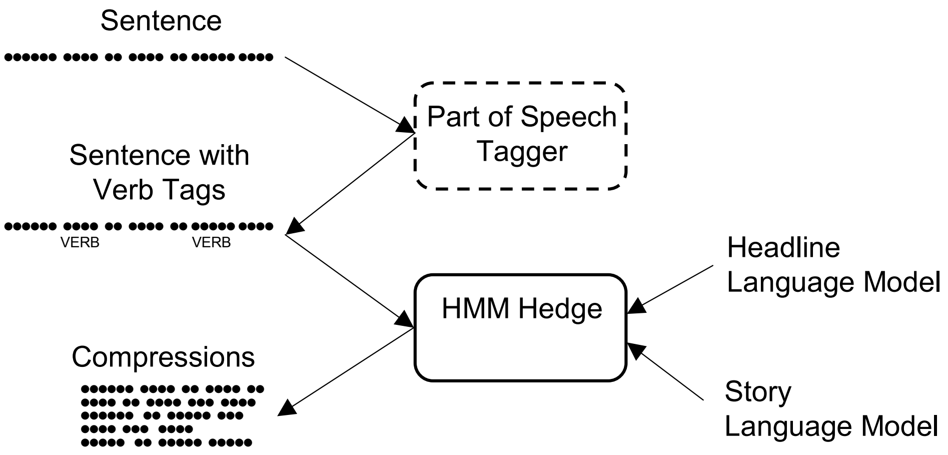


Figure 1.2: HMM Hedge architecture

ate on sentences or arbitrary blocks of contiguous words, however best results have been achieved by taking advantage of sentence boundaries. Pre-processing for HMM Hedge includes tagging verbs so that compressions can be forced to contain at least one verb. Part-of-speech tagging was done using TreeTagger[3] (Schmid, 1994). The system output is a set of sentence compressions associated with compression-specific features. HMM Hedge is presented in Chapter 3.

The architecture of the Trimmer compression tool is shown in Figure 1.3. Documents are preprocessed in two ways. The named entities and time expressions are tagged to produce a set of entity tags and a parser is used to generate a parse tree for each sentence in the document. Currently, Charniak's parser is used, but any parser compatible with the Penn Treebank conventions can be used. BBN's IdentiFinder$^{\text{TM}}$ is used for entity tagging. IdentiFinder is discussed in more detail in Section 4.2.2. Trimmer integrates the entity and parse information, and uses

as part of this dissertation and modules with dotted lines represent existing software tools used as supporting technology.

[3] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

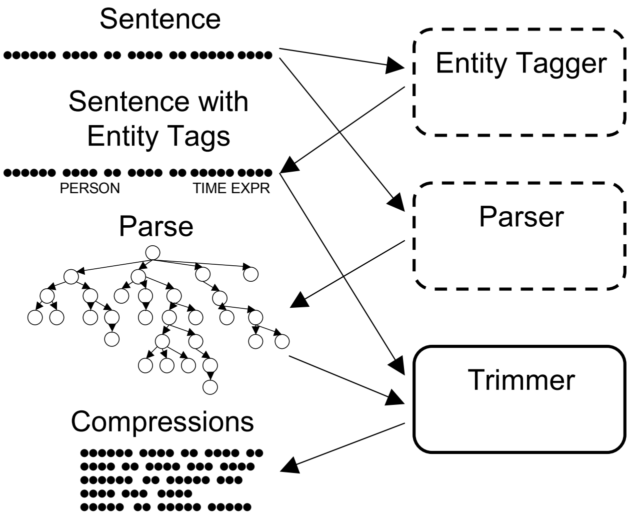


Figure 1.3: Trimmer architecture

syntactic shortening rules to generate a set of candidates and trimmer features, such as how many rules were applied, for each sentence. Trimmer will be presented in Chapter 4.

Topiary generates candidates using a combination of sentence compression and topic term discovery. The Topiary architecture is shown in Figure 1.4. Topic terms are automatically discovered based on a corpus of documents and assigned to the specific documents from which the sentences are drawn using topic discovery, Unsupervised Topic Discovery, and topic assignment, OnTopic$^{\mathrm{TM}}$, tools developed by BBN. UTD and OnTopic are discussed in Section 4.3.1. Topiary uses Trimmer to compress the sentences and combines the compressions with topic terms to produce candidates. The Topiary candidates are associated with Topiary-specific features, such as number of topic terms, as well as the Trimmer-specific features. Topiary will be presented in Section 4.3.

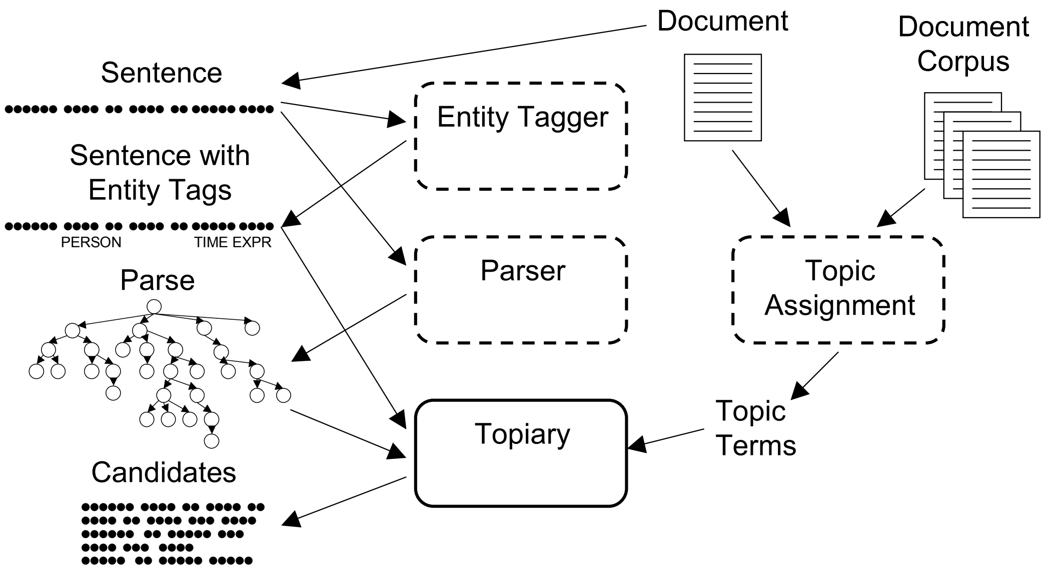


Figure 1.4: Topiary architecture

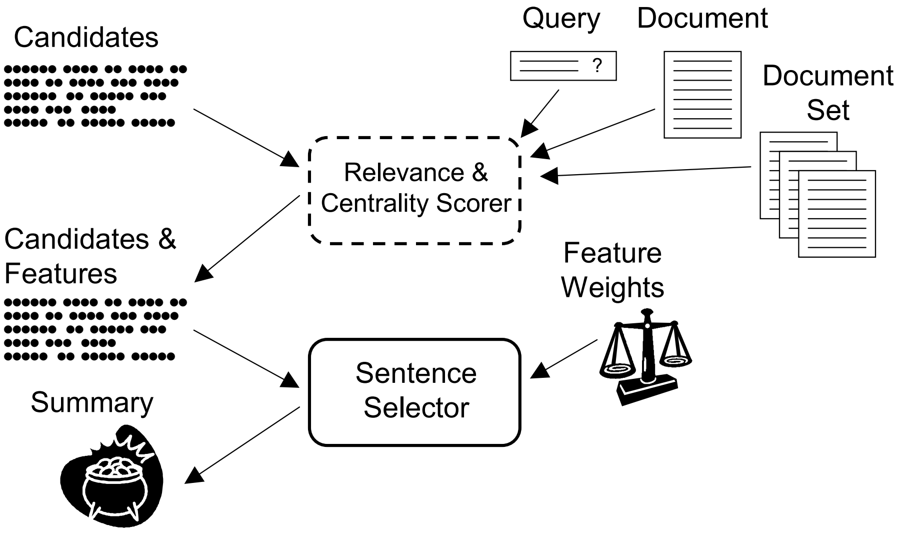


Figure 1.5: Selector architecture

The sentence selector, shown in Figure 1.5 chooses the best candidate or candidates to include in the final summary based on linear combinations of features. Some of the features come from direct observation of the candidates, such as length. Others features are products of the compression process, specific to HMM Hedge, Trimmer, or Topiary. Finally, additional features are calculated based on the relevance and centrality of the candidate and the document from which it was extracted. Relevance is the similarity of a candidate to a query. Centrality is a measure of how central a candidate is to the document in which it occurs. These features are calculated using the Uniform Retrieval Architecture (URA), and will be described in Section 5.3.1. The selector uses optimized feature weights to select the highest scoring candidate or candidates. The sentence selector for single document summarization will be introduced in Section 4.5.

The feature weights used by the sentence selector are learned by producing candidates with features from a training corpus and scoring each candidate with an automatic summarization evaluation tool. The automatic evaluation tool, Rouge, is described in Section 2.3.1. The weights on the features are then tuned using machine learning to select candidates that optimize the automatic evaluation score over the corpus. The tuning is done using BBN's Optimizer, described in Section 3.3.

For short, single-document summaries that consist of a single sentence, the operation of the sentence selector is straightforward. The summary consists of the highest-scoring candidate. In the work presented here, single-document summaries are all short, single-sentence headlines.[4] For longer summaries that contain multiple

[4] Although the generation of multi-sentence, single document summaries is a worthwhile task,

sentences, such as summaries of multiple documents, the process is more complex because relevance and centrality must be balanced with anti-redundancy. The MASC framework applied to a multi-document summarization task will be referred to as Multi-Document (MuD) MASC. The sentence selector for Mud MASC is an adaptation of Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998). A summary is constructed by iteratively selecting the highest scoring candidate until the desired summary length is reached. After each iteration dynamic features of the candidates, such as redundancy to the current summary state, are recalculated. MuD MASC is presented in Chapter 5 and the sentence selector for MuD MASC is presented in Section 5.3.

This section has given a brief overview of the MASC framework and the filtering, compression and selection tools that have been developed within the framework. In the remainder of the dissertation these tools will be described in greater detail. Evaluations and experiments within the MASC framework will demonstrate the potential of sentence compression as a component in an automatic summarization system.

1.2 Contributions

This dissertation yields the following contributions:

- The demonstration of the usefulness and potential of sentence compression for a variety automatic summarization tasks,

the application of MASC to this problem remains as future work.

- The demonstration through human studies that fluent and informative summaries can be constructed for written news stories by selecting words in order from the story, and

- The demonstration of the usefulness and potential of multiple candidate compression tools combined with feature-based candidate selection tools for a variety of automatic summarization tasks,

- The development of a framework for automatic summarization based on filtering, sentence compression and candidate selection, covering single-document and multi-document summarization tasks,

- The use of Headlinese, a compressed version of English in which newspaper headlines are written, as a model for very short, informative summaries using both a statistical sentence compression tool and a syntactic compression tool,

- The demonstration that Topiary, a system that combines topic terms with compressed text performs better than topic terms alone, sentence compression alone, and simple baselines in a variety of summarization tasks,

- The adaptation of automatic summarization techniques to novel genres, including broadcast news transcripts, email threads and cross-language summarization.

1.3 Dissertation Outline

The next chapter will relate the contributions of this dissertation to other work in Automatic Summarization and related Natural Language Processing fields. Chapters 3 and 4 will introduce two implementations of sentence compression, HMM Hedge and Trimmer, and discuss their use within the MASC framework for single-document summarization. Chapter 4 will also discuss Topiary, a candidate generation system that combines topic terms with fluent text from Trimmer. Chapter 5 describes multi-document summarization using the MASC framework, using both HMM Hedge and Trimmer as compression tools. The application of these summarization techniques to novel genres is discussed in Chapter 6. Finally, Chapter 7 summarizes (manually!) the dissertation and suggests areas for follow-on work. The appendices give implementation-level details of the major original software tools presented in the dissertation.

Chapter 2

Background and Related Work

This chapter presents an overview of the problem of text summarization, automatic summarization systems, and summarization evaluation. Some of the key enabling technologies are presented, e.g., sentence selection and sentence compression. In addition, the challenges faced by existing summarization systems are presented, and those addressed in this dissertation are identified.

2.1 Text Summarization

Text summarization, as a task performed by humans, involves reading and understanding a document for content, then generating a new document expressing a concise version of the content. A summary will not contain all the information from the source, but presents only the most important information. Automatic text summarization is the generation by a machine of output that presents important information from a source document in a compressed form, and in a manner that meets a user's or application's needs (Mani et al., 2002).

Newspaper headlines are a natural example of human summarization. Headlines are written by copy editors after an article is complete. The copy editors try to construct headlines that satisfy three goals: to summarize the story, to draw in the reader and to fit in the specified space (Rooney and Witte, 2000). In order to achieve these goals, headline writers adopt a form of compressed English, sometimes referred to as Headlinese (Mårdh, 1980). Headlinese is designed to convey the most information in the least space, while remaining comprehensible to untrained readers. This makes it an attractive model for automatic summarization systems. Some differences between Headlinese and standard English are the omission of determiners and forms of the verb "to be", and use of present tense for events in the past.

Consider the following headlines.

(3) (i) Under God Under Fire
(ii) Pledge of Allegiance
(iii) U.S. Court Decides Pledge of Allegiance Unconstitutional

Headline (3i) is an *eye-catcher*, designed to make the reader curious enough to read the article in order to find out what is going on, hopefully creating an information need so compelling that the reader purchases a newspaper. A reader could guess what happens in the article if he or she already knew that a court was considering the constitutionality of the pledge of allegiance. Headline (3ii) tells the reader that the topic is the pledge of allegiance, but does not tell what happened. Headline (3iii) does tell what happened. This illustrates the difference between *indicative* summaries, which identify topics and help a reader determine if a document is relevant to an information need, and *informative* summaries, which can stand as surrogates for the document in satisfying an information need. HMM Hedge and Trimmer aim to produce informative summaries. Topiary combines informative and indicative elements.

A further distinction is drawn between *abstractive* and *extractive* summaries.

Abstractive summaries, or *abstracts* are newly written content not found in the source document. Newspaper headlines, book reports and TV Guide program descriptions are examples of human-written abstractive summaries. Extractive summaries, or *extracts* are drawn entirely or in part from the document. Lead sentences of newspaper stories are extractive summaries. The systems described in this dissertation are extractive, but modify the extracts to create new text. Topiary is partly abstractive because the topic terms do not necessarily occur in the document.

Summaries can be generic or focused for a query or user profile. In a generic summary, information content is determined to be important based on its prominence within the document. In summaries that are focused toward a query or user profile, information about the user's information need or about the general interests of the user determine which information in the document is important enough to appear in the summary. The summarization systems presented in this dissertation are query-focused, but can create generic summaries in situations where no query exists.

Two commonly recognized summarization tasks are *single document summarization* and *multi-document summarization.* Single document summarization is commonly used for relevance judgment and question answering tasks. Multi-document summarization is performed on a set of related documents, with the goal of providing an overview of the entire document set. The relationship among the documents can include:

- following a single event over a short time, such as a flood in the Yangtze Valley,
- following a single event over a long time, such as negotiation of power sharing in Cambodia,

- multiple similar events, such as firings of CEOs,
- discussion of an issue, such as gun control, or
- documents on different topics from a single source, such as an issue of a newspaper or a program of broadcast news stories.

Issues of redundancy and coherence become important in multi-document summarization, because documents might repeat important information, and sentences from different documents might be put together outside of their original context.

A compression ratio is a way of expressing the degree of summarization required. Mani and Bloedorn (1999) show that informative summaries perform better at higher compression ratios of about 35% - 40%, because at higher compression ratios, summaries can include more information from throughout the document. Jing et al. (1998) shows that evaluation results for a system can differ greatly at different compression ratios. Alternately, a hard limit in characters or words can be specified regardless of the length of the source document. The systems described in this dissertation have used hard size limits, so the compression ratio depends on the size of the document. For the single document summarization this can range from 2% to 20%, and for multi-document summarization from 2% to 5%.

2.2 Automatic Summarization Systems

The first efforts at automatic text summarization consisted of selecting important sentences from a document and concatenating them together. Luhn (1958) uses term frequencies to measure sentence relevance. Sentences are included in the summary if the words in the sentence have high enough term frequencies. Luhn used a stoplist to exclude common words with little topic-specific value, such as prepositions and determiners, and also aggregated terms by orthographic similarity. Salton (1988) observed that documents in a certain domain will share certain common words beyond the obvious stop words. Terms that are common for the domain will have high term frequency in all documents about that topic, and thus are not good topic-sentence indicators. The relevance of a term in a document is inversely proportional to the number of documents in the collection containing the term. For example, documents about classical music will often contain the terms *classical*, *music* and *composer*, and so these terms are not good indicators of topic within the classical music domain. A measure which takes both term frequency within a document and term rarity in the general collection is *tf.idf*. tf_i is the frequency of term i in a document and idf_i is the inverted document frequency where

$$idf_i = log(\frac{N}{df_i}),$$

N is the number of documents in the collection, and df_i is the number of documents containing term i.

Systems that are constrained to select sentences from the document or document set are limited in several ways. First, they are limited to sentences that actually occur. Highly relevant sentences might also contain non-relevant or redundant material. Second, if the compression ratio or hard length limit is very small, the system might not be able to produce output that meets the length constraint. Both problems can be addressed by sentence compression, also called sentence shortening, sentence condensation, sentence reduction, and sentence simplification.

2.2.1 Sentence Compression in Automatic Summarization

One of the earliest automatic summarization systems to use sentence compression was Grefenstette (1998) in which a rule-based approach was used to help blind readers using text-to-speech software skim a page of text. Several levels of compression were used, the most severe of which left only proper nouns.

Syntactic approaches to compression have been used in single-document summarization systems such as Cut-and-Paste (Jing, 2000) and also in multi-document summarization systems such as SC (Blair-Goldensohn et al., 2004) and CLASSY (Conroy et al., 2005, 2006b). The SC system pre-processes input to remove appositives and relative clauses. CLASSY uses an HMM sentence selection approach combined with a conservative sentence compression method based on shallow parsing to detect lexical cues to trigger phrase eliminations.

Daumé and Marcu (2005) use syntactic compression as a component of a Bayesian query-focused summarization system. Four syntactic trimming rules are available. Different subsets of these rules can be applied as post-processing to all sentences uniformly. Siddharthan et al. (2004) remove appositives and non-restrictive relative clauses from sentences before submitting them to a clustering component, and find that this improves the quality of the clusters. Melli et al. (2006) apply summarization rules that aim to remove from the sentence only the content that deals with local document context, and nothing that could be part of an answer to a query.

Jing and McKeown (2000) present an extractive system in which selected sentences are post-processed by syntactic trimming. Sentences are parsed by a syntactic parser that also annotates thematic roles. For each node in the tree a decision is made about whether that node should be trimmed. This decision is based on a combination of grammatical, context and corpus considerations. Each node is determined to be grammatically obligatory or not with respect to its parent. A lexicon, with entry-specific information about obligatory arguments, is used to make this decision. Words are regarded as important to the main topic of discussion if they are linked via WordNet (Miller et al., 1993) or morphologically to other words in the local context. Finally, a corpus of sentence compressions by human professionals was analyzed by a decomposition program (Jing and McKeown, 1999) to match phrases from the source document to the human summaries. A constituent is removed from the tree if it is not grammatically obligatory, is not thematically important, and has high likelihood of being removed by a human.

All of these syntactic compression systems share the characteristic that a single sentence compression method is applied to all sentences, and a single compression is produced for each sentence. Trimmer differs from them in that multiple compressed candidates of each sentence are generated. The potential of multiple alternative compressions has also been explored by Vanderwende et al. (2006).

2.2.2 Topic Terms in Automatic Summarization

Summaries may also contain lists of words or short phrases that denote important topics or concepts in the document. In particular, extractive topic summaries consist of keywords or key phrases that occur in the document. Topiary is an extension to Trimmer that constructs headlines combining compressed sentences with topic terms. This approach is similar to the work of Euler (2002), except that Euler uses topic lists to guide sentence selection and compression toward a query-specific summary, whereas Topiary uses topics to augment the concept coverage of a generic summary.

In contrast to Topiary, which combines sentence compression with topic terms, others have constructed summaries directly from topic terms. For example, Bergler et al. (2003) choose noun phrases that represent the most important entities as determined by noun phrase coreference chains. Wang et al. (2005) propose a baseline system that constructs headlines from topic descriptors identified using term frequency counts; this system was reported to outperform LexTrim, their independently implemented Topiary-style system. Zhou and Hovy (2003) construct fluent summaries from a topic list by finding phrase clusters early in the text that contain important topic words found throughout the text.

The task of assigning topic terms to documents is related to text categorization, in which documents are assigned to pre-defined categories. The categories can be labeled with topic terms, so that the decision to put a document in a category is equivalent to assigning that category's label to the document. Assigning topic terms to documents by categorization permits the assignment of terms that do not occur in the document. Lewis (1999) describes a probabilistic feature-based method for assigning Reuters topics to news stories. Along these lines, OnTopic™ (Schwartz et al., 1997) uses an HMM to assign the topics to a document.

2.2.3 Noisy Channel Model in Automatic Summarization

Knight and Marcu (2000, 2002) introduce a statistical sentence compression system based on the noisy-channel model. The noisy-channel approach has been used for a wide range of Natural Language Processing (NLP) applications including speech recognition (Bahl et al., 1983); machine translation (Brown et al., 1990); spelling correction (Mays et al., 1990); language identification (Dunning, 1994); and part-of-speech tagging (Cutting et al., 1992). In the context of automatic summarization, the underlying intuition of the noisy channel model is that the observed signal, in this case a sentence, has been altered from an unseen source, the compressed sentence, by transmission through a channel that adds words.

In Knight and Marcu (2000, 2002) there are three components:

- A source model of the original short string
- A channel model of how short strings are expanded into long ones
- A decoder that searches for the most likely short string that accounts for the observed long string

The probabilities in Knight's work apply to parse trees, and syntactic rules that grow the observed parse tree from the compressed parse tree. Banko et al. (2000) and Turner and Charniak (2005) also use the noisy channel model to compress sentences for summarization.

Like other summarization systems based on the noisy-channel model, HMM Hedge treats the observed data (the story) as the result of unobserved data (headlines) that have been distorted by transmission through a noisy channel. The effect of the noisy channel is to add story words between the headline words. HMM Hedge differs from Knight and Marcu (2000), Banko et al. (2000), and Turner and Charniak (2005), in that it does not require a corpus of paired stories and summaries. HMM Hedge uses distinct language models of news stories and headlines, but does not require explicit pairings of stories and summaries.

2.2.4 Multi-Sentence Summaries

Compression tools within the MASC framework generate multiple candidate compressions for both single and multi-document summarization. For single document summarization it is necessary to select a compressed candidate to serve as the summary. Multi-document summaries typically contain multiple compressed candidates in a summary. In this case, the selection phase consists of selecting a set of candidates that maximize coverage while minimizing redundancy. A common approach is to rank candidate sentences according to a set of features and iteratively build the summary, appropriately re-ranking the candidates at each step to avoid redundancy. MEAD (Radev et al., 2004) scores source sentences according to a linear combination of features including centroid, position and first-sentence overlap. These scores are then refined to consider cross-sentence dependencies, such as redundancy, chronological order and source preferences. The MASC framework differs in that multiple variants of a single source sentence are available to the sentence selector.

Minimization of redundancy is an important element of a multi-document summarization system. Carbonell and Goldstein (1998) propose a technique called

Maximal Marginal Relevance (MMR) for ranking documents returned by an information retrieval system so that the front of the ranked list will contain diversity as well as high relevance. Goldstein et al. (2000) extend MMR to multi-document summarization. The MASC sentence selector borrows the ranking approach of MMR, but uses a different set of features. Like MEAD, these approaches use feature weights that were optimized to maximize an automatic metric on training data.

2.2.5 Novel Genres and Applications for Summarization

Several researchers have shown the importance of summarization in domains other than written news. These novel domains include voice mails (Koumpis and Renals, 2000), multi-party dialogs (Zechner, 2002), newsgroups (Newman and Blitzer, 2003), and blogs (Zhou and Hovy, 2006). Particular interest has been paid to summarization of broadcast news and emails.

Broadcast news differs from written news in style. For example, Christensen et al. (2004) distinguishes between *read* news, which is pre-planned, and *spontaneous* news, such as interviews and panel discussions. Most approaches to broadcast news summarization use Automatic Speech Recognition (ASR) as a preprocessing step. Systems exist that summarize broadcast news by sentence extraction (Maskey and Hirschberg, 2003; Christensen et al., 2004) and by topic term lists (Jin and Hauptmann, 2001). Document and sentence boundary detection are important preprocessing steps: Christensen et al. (2005) present utterance (sentence) and topic (document) segmentation of broadcast news programs as pre-processing stages for summarization. Maskey and Hirschberg (2003) treat summarization of an entire news program as a multi-document summarization problem and use speaker turns as the unit of extraction. It has also been proposed that broadcast news summarization can be achieved without ASR, solely using acoustic/prosodic features of the audio and structural features of broadcast news programs (Maskey and Hirschberg, 2005).

Sentence compression has been applied to broadcast news (Hori et al., 2002). HMM Hedge is similar to Hori et al. (2002) in that a dynamic programming technique is used to compress sentences by selecting a subset of words from the source, and that multiple compressions are considered. However, HMM Hedge is different in that compressions are generated independently at the sentence level, biases are used to mimic Headlinese, and a separate feature-based selection stage is used to select the best compressed candidates.

Automatic summarization has also been applied to the genre of email messages and email threads. Potential uses for email summarization tools are suggested by Stuff I've Seen (Dumais et al., 2003), a tool for indexing a variety of computer information formats including email queues, and the email search tasks evaluated in TREC-2005 (Craswell et al., 2005). Previous work has employed a corpus of emails sent among the board members of the ACM chapter at Columbia University (Rambow et al., 2004). Researchers have also examined summarization of archived discussion lists (Nenkova and Bagga, 2003; Newman and Blitzer, 2003; Wan and McKeown, 2004), email gisting by means of noun-phrase extraction (Muresan et al., 2001), thread-driven email summarization (Lam et al., 2002). However, the work presented in this dissertation is unique in the application of sentence compression to email thread summarization.

Cross-language summarization is the generation of a summary in a language that is different from the language (or languages) of the source documents. In the field of Cross-Language Information Retrieval (CLIR) summarization is used to create surrogates for the retrieved source documents in the target language. The IR task determines the desired characteristics of the summarization. For example, a relevance judgment task would require an indicative summary (Oard et al., 2004) while a question-answering task would require an informative summary (He et al., 2004). Machine translation is commonly used as a pre-processing step (Oard et al., 2004; He et al., 2004; Lim et al., 2004), but other surrogates include color-coded thumbnails showing the positions and identities of search terms (Ogden et al., 1999) and noun phrases (López-Ostenero et al., 2005). The application of Trimmer to cross-language summarization is similar to Oard et al. (2004) in that it applies post-processing to Machine Translation output. However the application of HMM Hedge is distinctive in that translation and summarization are achieved in the same process.

2.3 Evaluation of Text Summarization

In developing summarization systems it is important to have tools for measuring how system changes affect performance. This section will describe some of the evaluation methods used within the community at formal evaluations such as the

Document Understand Conferences (DUC) (Dang and Harman, 2006), SUMMAC (Mani et al., 2002) and the Multilingual Summarization Evaluation.[1] Versions of HMM Hedge, Trimmer and Topiary have been submitted to these conferences from 2002 through 2006, as described in Sections 3.4, 4.6, 5.4.4 and 5.5.6.

There are two kinds of summarization evaluations: *intrinsic* and *extrinsic*. Intrinsic evaluations are based on direct observations of the system output. Human intrinsic evaluations measure clarity, coherence, fluency, and informativeness (Jing et al., 1998). Automatic intrinsic evaluations compare system outputs to model summaries, written by humans. Extrinsic evaluations measure the performance on some other task (human or automatic) in which summarization system outputs affect the task. Execution of instructions, information retrieval, question answering and relevance assessments are examples of extrinsic evaluations. (Mani, 2001)

2.3.1 Automatic Intrinsic Evaluation of Summarization

Three intrinsic evaluation tools that have been been widely used in the NLP community are Bleu, Rouge, and Pyramid. All three make use of human-generated reference summaries. Bleu and Rouge are fully automatic and Pyramid is semi-automatic. Bleu is primarily a Machine Translation (MT) metric, Rouge and Pyramid are summarization metrics.

Bilingual Language Evaluation Understudy (Bleu) (Papineni et al., 2002) is an n-gram-based precision evaluation metric for MT. The Bleu precision metric is the percentage of n-grams in a candidate summary that match n-grams in the refer-

[1] http://projects.ldc.upenn.edu/MSE

ence summaries. Modifications to the metric correct for two MT phenomena. First, matching n-grams count only once per reference. This avoids giving high scores to translations composed entirely of high frequency, low content words. Second, a brevity penalty is applied if the candidate is shorter than the references.

Bleu is a precision measure because it measures how much of the content that appears in a candidate translation should properly appear in an ideal translation. As such it rewards systems that rarely say the wrong thing, but doesn't penalize systems that fail to say the right things. For this reason, Bleu has not been widely used for evaluation of summarization. As a precision measure it rewards systems that rarely say the wrong thing, but does not penalize systems that fail to say the right thing.

As a response to Bleu's limitations with respect to evaluation of summarization, University of Southern California's Information Sciences Institute (ISI) proposed the recall-based metric Recall Oriented Understudy of Gisting Evaluation (Rouge) (Lin, 2004). Rouge calculates the word n-gram overlap of a summary submitted for evaluation (peer) with a set of human-generated summaries (models). The n-gram overlap of a peer with the models is the number of n-grams in the peer that also occur in the models. For a single summary, the recall is the percentage of n-grams in the models that also occur in the peer. The precision is the percentage of n-grams in the peers that also occur in the models. For a system, the average recall and precision are the averages over all the summaries in the test set. Thus, Rouge-1 average recall refers to the average over all the document clusters of the unigram recall, Rouge-2 average precision refers to the average over the document clusters of the bigram precision, etc. ROUGE awards a brevity bonus because concise summaries without non-relevant material are preferred.

Four different ROUGE scores are available.

- ROUGE-N counts contiguous n-grams, where n ranges from 1 to 4.
- ROUGE-L is a combination of recall, precision and longest common subsequence.
- ROUGE-W is like ROUGE-L but uses a weighting factor for longest number of consecutive matching words.
- ROUGE-S uses skip bigrams: pairs of words in sentence order, ignoring gaps.

ROUGE can be configured to use jackknifing, which allows the authors of the model references to be compared to the peers. Suppose there are four model summaries for each document, a typical number. Jackknifing means that rather than comparing each peer to all four models, each peer is compared three times to three three-member subsets of the four models. In this way the human authors can be compared to the peers by scoring each author's summaries with respect to the other three models.

High correlations with human quality judgments have been shown for ROUGE-1 Recall in the case of 75-character single-document summaries, and for ROUGE-2 recall for 100-word single document summaries and multi-document summaries. These metrics have been widely used in the NLP community for automatics summarization evaluation. In addition, they are suitable for use in system development and testing because they do not require human involvement beyond the creation of the model references. For these reasons, the automatic evaluations in this dissertation will use ROUGE-1 recall for single-document summarization and ROUGE-2 recall for multi-document summarization.

The Pyramid Method (Nenkova and Passonneau, 2004; Passonneau and Nenkova, 2003) is semi-automatic in that manual annotation of reference summaries and candidate summaries is required, but the scoring is automatic. Reference summaries are annotated for Semantic Content Units (SCU). There is not a formal definition for SCU, but intuitively SCUs are single facts at the clause level. For example, the information that some Libyans were indicted for the Lockerbie bombing is a separate SCU from the information that the indictment took place in 1991. The SCUs are given weights equal to the number of reference summaries in which they occur, and organized into a pyramid. A highly central SCU that occurs in every reference will have a score equal to the total number of references, and will appear in the top tier of the pyramid. A SCU that occurs in only one reference will get a score of 1 and appear in the bottom tier of the pyramid.

Pyramid candidates are also manually annotated for SCUs. A candidate of a particular size can achieve an optimal score if all its SCUs are as high up in the pyramid as possible. The score is the ratio of the sum of the weights of a candidate's SCUs to the sum of the weights of an optimal candidate of the same size.

The Pyramid Method has the advantage that it is based on semantic content rather than n-gram matching. However the scoring method is highly labor intensive, and thus unavailable to system developers as a tool for measuring incremental system improvements.

2.3.2 Human Extrinsic Evaluation

In order to be regarded as useful, a summary should provide enough information to serve as a surrogate for making decisions or taking actions. An extrinsic evaluation measures subjects' performance of a task in which they make use of automatically generated summaries from different sources. In designing an extrinsic task it is important that the task be unambiguous enough that subjects can perform it with a high level of agreement. If the task is so difficult that subjects cannot perform with a high level of agreement – even when they are shown the entire document – it will not be possible to detect significant differences among different summarization methods because the amount of variation due to noise will overshadow the variation due to summarization method. Question answering, information retrieval, and relevance judgment have been used as extrinsic tasks.

In a relevance judgment task, subjects are asked to decide whether a document is relevant to a query or topic based only on having seen a summary of the document. An alternative to relevance judgments with gold standards is proposed in Dorr et al. (2005). Relevance prediction measures whether a subject is able to predict his or her own relevance judgment on a full text document by first looking at a summary of the document. This approach was found to be a more reliable measure than gold standard evaluations, and was able to support stronger statistical statements about the benefits of summarization.

2.3.3 Correlation of Automatic and Human Evaluation Tools

Ultimately the value of an automatic evaluation tool is how well it serves as a surrogate for a direct measure of system success. In the case of automatic summarization this would be a measure of how well system output serves a user's or application's information need. In this respect automatic evaluation of summarization remains an area of active research. High correlations with human measures have been claimed for Rouge (Lin and Hovy, 2003). The Pyramid Method has been shown to have high inter-annotator reliability (Nenkova and Passonneau, 2004). Relevance prediction is proposed by Dorr et al. (2005); Hobson et al. (2007) as an extrinsic task to which automatic summarization evaluation tools can test their correlation. The automatic evaluations presented in this dissertation use Rouge because it has been accepted as the official evaluation tool of the Document Understanding Conference evaluations since 2003. In addition, a human evaluation based of HMM Hedge, Trimmer, and Topiary using relevance prediction, will be presented in Section 4.7.

2.4 Discussion

Extractive summarization systems that select from only sentences that occur in documents or a single compression of sentences in the documents are less able to avoid problems of relevance and redundancy. Sentence compression is a powerful tool for addressing this limitation. However, no compression tool that produces a single output for each sentence can produce the best candidate for every sentence in every context. For example, the relevance of the information in a candidate sentence or sentence compression depends on the user's information need and the centrality of a candidate depends on the document or document set in which it appears. Similarly, a candidate's redundancy depends on the content of the partially completed multi-sentence summary into which the candidate is to be added.

One possible solution to this problem is to perform sentence compression as a post-processing step after sentence selection. However the problem remains that even a query-focused compression algorithm that produces a single compression might not generate the best compression, or set of compressions in a multi-sentence summary.

The work presented in this dissertation addresses this problem by introducing Multiple Alternative Sentence Compressions (MASC) framework, which makes it possible for sentence selectors to choose among different compressions of source sentences. The candidates generated by MASC contain different subsets of the information contained in the original source sentences. The expanded pool of candidates enables sentence selectors to construct summaries from extracts that contain less information that is redundant, non-relevant or non-central. Recently Vanderwende et al. (2006) have also begun to explore the use of multiple compressions.

Saving space is an important task in summary generation. Systems that limit their output to sentences that are grammatically well formed in general English devote space to elements that are not necessary for comprehension. This is particularly vital when the space is fixed and very small. Writers of newspaper headlines use a compressed variety of English called Headlinese, that saves space by using constructions that are not grammatical in general English, but which are easily understood by readers. Another contribution of this dissertation is modeling sentence compression to produce output that mimics Headlinese.

Sentences taken out of context can often tell what happened but lack enough background information to satisfy a user's need. This can be due to pronoun references, or an incomplete mention of an entity that was identified more fully in a previous sentence. One approach is to select clumps of contiguous sentences. However in automatic headline generation, there is generally room for only one sentence. The contribution of this dissertation with respect to this problem is Topiary, a headline generation system that combines compressed fluent text for informativeness with topic terms for informativeness.

Chapter 3

HMM Hedge Sentence Compression

Underlying all the sentence compression tools discussed in this dissertation is the idea that a fluent, informative summary for a document can be constructed by selecting words in order from the document. Originally this was intended to mimic the style of newspaper headlines, performing the task of headline generation. Feasibility studies showed that humans could almost always successfully construct headlines by selecting words in order from documents, without respect for sentence boundaries and allowing some morphological variation.

The first section of this chapter will describe two feasibility studies that tested this general approach to headline generation. Subsequent sections present HMM Hedge, a statistics-based sentence compression tool, with a focus on the linguistically-motivated modifications that help the system mimic Headlinese. HMM Hedge, shown in Figure 3.1, is demonstrated in use within the MASC framework for the task of generating very short, single-document summaries. Automatic evaluations of HMM Hedge as a component of an automatic summarization system explore the effects of MASC's filtering stage, generation of single or multiple compressions, and selector feature weight training on system performance. Section 5.4.1 will present HMM Hedge applied to multi-document summarization within the MASC framework. An implementation-level description of the HMM Hedge compression tool is

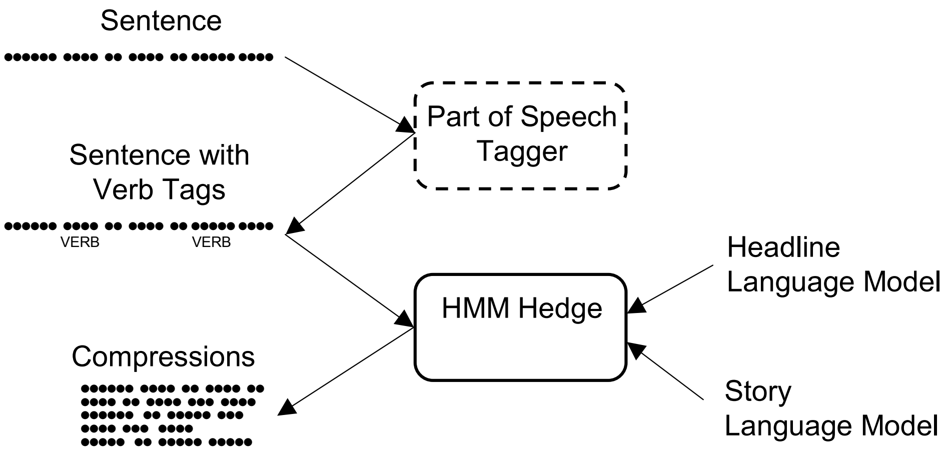


Figure 3.1: HMM Hedge architecture

given in Appendix A.

3.1 Feasibility Study

Two feasibility studies were conducted to determine whether it was possible for humans to construct very short informative summaries in the style of a newspaper headline for written news stories. The subjects were instructed to select words in order from the text. Another goal of the studies was to discover patterns in the human performance that could guide the development of an automatic summarization system.

In the first study three participants were shown 56 AP stories from 1989 from the TIPSTER corpus[1] and asked to construct headlines for the stories by selecting words from the stories while preserving the order that the words occurred in the stories. The participants were not given any instructions about whether or not to respect sentence boundaries in summary word selection. It was possible to construct

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93T3A

headlines for 53 of the stories. The remaining three stories were a list of commodity prices, a chronology of events and a list of entertainment events. The subjects demonstrated that the underlying idea of selecting words in order from documents is a feasible way to generate headlines.

The participants showed a bias for selecting words from near the front of the stories. Summaries for only 7 of the 53 stories used words beyond the 60th story word, and only one went beyond the 200th story word. The preference of subjects for words early in the document agrees the observation in Wasson (1998) that constraining Boolean queries of news documents to leading text improves precision in retrieval results.

In the second study two participants performed the same task on 73 AP stories from January 1, 1989 from the TIPSTER corpus but were allowed to use morphological variations of verbs in the source documents. For example, one participant constructed the summary in Sentence (4ii) from the source in Sentence (4i). Note that in the summary, *broke* has been replaced by *breaks*.

(4) (i) An earthen dike broke early Sunday, forcing the evacuation of an estimated 1,500 people for several hours and closure of a major interstate highway, authorities said.

(ii) earthen dike breaks forcing evacuation of 1,500 people and closure of highway

Of the 146 summaries written by the participants in this study, 2 did not meet the select-words-in-order constraint because of accidental word reordering. The participants created at least one fluent and accurate headline meeting the criterion for each of the stories. The average length of the headlines was 10.76 words. Even though the participants were given no instructions regarding sentence boundaries, they constructed headlines entirely of words from the first sentence 80.1% of the time, and at a finer grain, 86.7% of the headline words were chosen in order from the first sentence. Only 8.9% of the 146 summaries used words beyond the first sentence, and none of the 144 summaries meeting the words-in-order constraint used words from beyond the 4th sentence.

This study also shows that summarization through selecting words in order from documents is a task which humans can perform and that humans have a bias for words early in the story. The second study additionally shows that humans have a bias for constructing headlines from words in a single sentence.

Consider the following lead sentence from a news story and corresponding compression of that sentence, which illustrate the general approach of compression by selecting words in order.

(5) After months of debate following the Sept. 11 terrorist hijackings, the Transportation Department has decided that airline **pilots** will **not** be **allowed to have guns in** the **cockpits**.

(6) Pilots not allowed to have guns in cockpits.

The bold words in (5) form a fluent and accurate headline, as shown in (6). This example will be referred to in subsequent sections.

3.2 HMM Hedge System Description

The intuition behind HMM Hedge is to treat the observed data (the story) as the result of unobserved data (headlines) that have been distorted by transmission through a noisy channel. The effect of the noisy channel is to add story words between the headline words. The model is biased by parameters to make the resulting headlines more like the observed language of newspaper headlines created by copy editors.

Formally, a story S is considered to be a sequence of N words. The goal is to find a headline H, a subsequence of words from S, that maximizes the likelihood that H generated the story S, or:

$$argmax_H P(H|S)$$

It is difficult to estimate $P(H|S)$ directly, but this probability can be expressed in terms of other probabilities that are easier to estimate, using Bayes' rule:

$$P(H|S) = \frac{P(S|H)P(H)}{P(S)}$$

Since the goal is to maximize this expression over H, and $P(S)$ is constant with respect to H, the denominator of the above expression can be omitted. Thus the goal is to find:

$$argmax_H P(S|H)P(H)$$

Let H be a headline consisting of words $h_1, h_2, ..., h_n$. Let the special symbols *start* and *end* represent the beginning and end of a headline, respectively. $P(H)$ can be estimated using a bigram model of Headlinese:

$$P(H) = P(h_1|start)P(h_2|h_1)...P(end|h_n)$$

The bigram probabilities of the words in the headline language were computed from a corpus of English headlines of 242,918 AP newswire stories from the TIPSTER corpus. The headlines contain 2,848,194 words from a vocabulary of 88,627 distinct words.

Given a story S and a headline H, the action of the noisy channel is to form S by adding non-headline words to H. Let G be the non-headline words added by the channel to the headline: $g_1, g_2, ..., g_m$. For the moment, assume that the headline words are transmitted through the channel with probability 1. An estimate is needed for $P(S|H)$, the probability that the channel added non-headline words G to headline H to form story S. This is accomplished using a unigram model of newspaper stories, which will be referred to as the general language, in contrast to the headline language. Let $P_{gl}(g)$ be the probability of non-headline word g in the general language, and $P_{ch}(h) = 1$ be the probability that headline word h is transmitted through the channel as story word h.

$$\begin{aligned} P(S|H) &= P_{gl}(g_1)P_{gl}(g_2)...P_{gl}(g_m)P_{ch}(h_1)P_{ch}(h_2)...P_{ch}(h_n) \\ &= P_{gl}(g_1)P_{gl}(g_2)...P_{gl}(g_m) \end{aligned}$$

The unigram probabilities of the words in the general language were computed from 242,918 English AP news stories in the TIPSTER corpus. The stories contain 135,150,288 words from a vocabulary of 428,633 distinct words.

The process by which the noisy channel generates a story from a headline can be represented by a Hidden Markov Model (HMM) (Baum, 1972). An HMM is a weighted finite-state automaton in which each state probabilistically emits a string. The simplest HMM for generating headlines consists of two states: an H state which emits words that occur in the headline, and a G state which emits all the other words in the story. This HMM is shown in Figure 3.2.

Since a bigram model of headlines is used, each state which emits headline

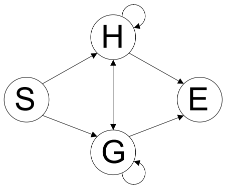


Figure 3.2: HMM for generating headlines from stories. S is the start state and E is the end state. The words of the story are emitted by the H and G states. The headline is composed of words emitted by the H state.

words must "remember" the previously emitted headline word. If headline words were not constrained to actually occur in the story, there would need to be an H state for each word in the headline vocabulary. However, because headline words are chosen from the story words, it is sufficient to have an H state for each story word. For any story, the HMM consists of a start state S, end state E, an H state for each word in the story, a corresponding G state for each H state, and a state G_{start} that emits words which occur before the first headline word in the story. An H state can emit only the word it represents. The corresponding G state remembers which word was emitted by its H state and can emit any word in the story language. A headline corresponds to a path through the HMM from S to E that emits all the words in the story in the correct order. Figure 3.3 shows the HMM for a story containing three words. In practice the HMM is constructed with states for only the first N words of the story, where N is a constant (60), or N is the number of words in the first sentence.[2]

In example (5), given earlier, the H states will emit the words in bold (pilots, not, allowed, to, have, guns, in, cockpits), and the G states will emit all the other

[2] Limiting consideration of headline words to the early part of the story is justified in Dorr et al. (2003b) where it was shown that more than half of the headline words are chosen from the first sentence of the story. Other methods for selecting the window of story words are possible and will be explored in future research.

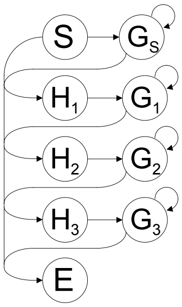


Figure 3.3: HMM for a Story Containing Three Words. S represents the start state and E the end state. The H states emit story words that appear in the headline and G states emit story words that do not appear in the headline. The G state associated with an H state emits all the words that occur in the story between that headline word and the next word in the headline.

words. The HMM will transition between the H and G states as needed to generate the words of the story. In the current example, the model will have states $Start, G_{start}, End$ and 28 H states with 28 corresponding G states.[3] The headline given in example (6) corresponds to the following sequence of states: *Start*, G_{start} 17 times, H_{pilots}, G_{pilots}, H_{not}, G_{not}, $\mathrm{H}_{allowed}$, H_{to}, H_{have}, H_{guns}, H_{in}, G_{in}, $\mathrm{H}_{cockpits}$, *End*. This path is not the only one that could generate the story in (5). Other possibilities are:

(7) (i) Transportation Department decided airline pilots not to have guns.

(ii) Months of the terrorist has to have cockpits.

Although (7i) and (7ii) are possible headlines for (5), the conditional probability of (7ii) given (5) will be lower than the conditional probability of (7i) given (5).

The Viterbi algorithm (Viterbi, 1967) is used to select the most likely headline for a given story. Length constraints are used to find the most likely headlines

[3] The subscript of a G state indicates the H state it is associated with, not the story word it emits. In the example, G_{pilots} emits story word *will*, G_{not} emits story word *be*, and G_{in} emits story word *the*.

consisting of W words, where W ranges from 5 to 15. Multiple backpointers are used so that the n most likely headlines at each length can be found.

3.2.1 Decoding Parameters

HMM Hedge is enhanced by three additional decoding parameters to help the system choose outputs that best mimic actual headlines: a position bias, a clump bias and a gap bias. The incorporation of these biases changes the score produced by the decoder from a probability to a relative desirability score. The three parameters were motivated by analysis of system output and their values were set by trial and error. A logical extension to this work would be to learn the best setting of these biases, e.g., through Expectation Maximization.

The position bias favors headlines which include words near the front of the story. This reflects the observations of human-constructed headlines, in which headline words tend to appear near the front of the story. The initial position bias p is a number between 0 and 1. The story word in the nth position is assigned a position bias of $log(p^n)$. When an H state emits a story word, the position bias is added to the desirability score. Thus, words farther along in the story carry a larger negative position bias than words near the front of the story. The smaller the position bias parameter, the stronger the bias is for early words. The current value of the position bias parameter is 0.95. Note that this generalization often does not hold in the case of human interest and sports stories, which may start with a hook to get the reader's attention, rather than a topic sentence.

Human-constructed headlines tend to contain contiguous blocks of story words. Example (6), given earlier, illustrates this with the string “allowed to have guns.” The string bias is used to favor “clumpiness,” i.e., the tendency to generate headlines composed of clumps of contiguous story words. The clump bias is a number between 0 and 1. The log of the clump bias is added to the desirability score with each transition from an H state to its associated G state or the end state. Transitions from an H state to its corresponding G state or the end state indicate the end of a clump, so the number of these transitions equals the number of contiguous blocks of story words in the headline. Headlines with many clumps carry a larger negative clumpiness bias than headlines with fewer clumps. The smaller the clumpiness bias parameter, the stronger the bias in favor of headlines with fewer but larger clumps of contiguous story words. The current value of the clumpiness bias parameter is 0.95.

The gap bias is used to disfavor headline “gappiness,” i.e., large gaps of non-headline words in the story between clumps of headline words. Although humans are capable of constructing fluent headlines by selecting widely spaced words, it was observed that HMM Hedge is more likely to combine unrelated material by doing this. At each transition from a G state to an H state, corresponding to the end of a sequence of non-headline words in the story, a gap bias is applied that increases with the size of the gap between the current headline and the last headline word to be emitted. This can also be seen as a penalty for spending too much time in one G state. The gappiness bias parameter is a number between 0 and 1. The smaller the gappiness bias parameter, the stronger the bias against headlines with large gaps.

The current values of the gappiness bias parameter are 0.99 for gaps of one word and 0.95 for gaps larger than one word.

Note that gappiness and clumpiness, although related are not actually the same property. Reducing the number of clumps also reduces the number of gaps between clumps. However gappiness is sensitive to the size of the gaps between clumps. Consider the following sentence and two distinct compressions.

(8) China's military has ordered soldiers to fight to the death to ensure that waterlogged dikes holding back the flooded Yangtze River do not collapse state media reported Monday.

(i) **China's military** has **ordered soldiers to fight to** the **death to ensure** that **waterlogged dikes holding back** the **flooded Yangtze River do not collapse** state media reported Monday.
China's military ordered soldiers to fight to death to ensure waterlogged dikes holding back flooded Yangtze River do not collapse.

(ii) **China's military** has ordered soldiers to **fight** to the death **to ensure** that waterlogged **dikes holding** back the flooded **Yangtze River do not collapse** state media reported Monday.
China's military fight to ensure dikes holding Yangtze River do not collapse.

Both compressions contain five clumps of contiguous story words. Sentence (8i) has 4 gaps of one word and 1 larger gap (the gap from *collapse* to the final period). Sentence (8ii) has no gaps of one word and 5 larger gaps.

An additional modification to the decoder is that each headline is constrained to contain at least one verb. That is to say, headlines which do not contain at least one verb are ignored, no matter how high the decoder score is. This also helps HMM Hedge mimic newspaper headlines, which typically contain a verb.

3.2.2 Morphological Variation for Verbs

One characteristic difference between newspaper headline text and newspaper story text is that headlines tend to be in present tense while story sentences tend to be in the past tense. Past tense verbs occur more rarely in the headline language than in the general language. HMM Hedge mimics this aspect of Headlinese by allowing morphological variation between headline verbs and the corresponding story verbs. Morphological variation for verbs is achieved by creating an H state for each available variant of a story verb. These H states still emit the story verb but they are labeled with the variant. HMM Hedge can generate a headline in which *proposes* is the unobserved headline word that emits the observed story word *proposed*, even though *proposes* does not occur in the story.

(9) (i) A group has proposed awarding $1 million in every general election to one randomly chosen voter.

(ii) Group proposes awarding $1 million to randomly chosen voter.

When an H state is labeled with a morphological variant of the story verb, the emit probability is no longer one. At present the probability that an H state labeled with a morphological variation of the story verb emits the story verb is $1/N$, where N is the number of morphological variants of the story verb represented in the H states. A future enhancement of HMM Hedge would be to train the emit probabilities to reflect the distribution of verb tenses in headlines and news stories.

3.2.3 Multiple Alternative Compressions with HMM Hedge

Multiple alternative compressions of a sentence may be generated with HMM Hedge. The Viterbi algorithm is capable of discovering n-best compressions of a window of story words and can be constrained to consider only paths that include a specific number of H states, corresponding to compressions that contain a specific number of words. For example, when HMM Hedge is configured for 5 best compressions at each length from 5-words to 15-words, it generates 55 compressions.

Table 3.1 shows the 5-best compressions of Sentence (8) with lengths (in words) of 5, 10 and 15.

3.3 Candidate Selection: HMM Hedge and Optimization

A linear combination of features is used to select among the multiple compressions provided by HMM Hedge. The features used with HMM Hedge are listed in Table 3.2.

The initial weights are based on the path cost calculations in the Viterbi decoder. The bigram, unigram and emit probabilities are combined with the decoding parameters to calculate the cost of a headline in the Viterbi decoder. This is augmented with a positive weight for headline length and a negative weight for the source sentence's position in the story.

BBN's Optimizer (Ostendorf et al., 1991; Schwartz et al., 1992), an implementation Powell's Method (Powell, 1965), is used to find the feature weights that maximize the ROUGE score over a set of training documents. Powell's Method is a

Five-word Compressions	soldiers held back Yangtze River
	soldiers holding back Yangtze River
	soldiers holding the Yangtze River
	soldiers fight back Yangtze River
	soldiers held the Yangtze River
Ten-word Compressions	China's military orders soldiers to fight back the Yangtze River
	military orders soldiers to fight to hold the Yangtze River
	military orders soldiers to fight to hold the Yangtze River
	China's ordered to fight to hold back the Yangtze River
	China's orders soldiers to fight to hold the Yangtze River
Fifteen-word Compressions	China's military orders soldiers to fight to death to hold the Yangtze River
	China's military orders soldiers to fight to death to ensure holds back the Yangtze River
	military orders soldiers to fight to death to hold the Yangtze River not state media
	China's military orders soldiers to fight to death to hold back the Yangtze River not
	China's military orders soldiers to fight to death to ensure held back the Yangtze River

Table 3.1: Examples of HMM Hedge sentence compressions

Feature	Initial	Optimized
Sum of story word positions of headline words	log(0.95) = -0.051	-0.046
Number of one-word gaps	log(0.99) = -0.010	4.193
Number of multi-word gaps	log(0.95) = -0.051	1.205
Number of clumps	log(0.95) = -0.051	-0.180
Story sentence position	-10.0	-18.005
Length in words	1.0	-3.409
Length in characters	1.0	2.360
Bigram probability in Headline Model of headline words	1.0	1.565
Unigram probability in General Model of non-headline words	1.0	0.977
Emit probability of headline words given story words	1.0	1.300

Table 3.2: Features and weights used for HMM candidate selection

technique for finding the minimum or maximum of a multi-variable function. In this case the Rouge score, expressed as the number of n-gram matches in a candidate divided by the number of n-grams in the models, are the known values for linear combinations of known sets of feature values. The optimization technique discovers the feature weights that produce the highest value for the sum over a set of documents of the number of the n-gram matches in a candidate divided by the sum over a set of documents of the model n-gram counts. Feature weights are optimized for Rouge-1 recall for short single document summarization, because this metric has been shown to have a good correlation with human judgment and is the accepted evaluation metric for this task in the summarization community (Lin, 2004; Harman, 2004).

HMM Hedge was run on 624 written news documents from the test data of the DUC2003 single-document summarization task. ROUGE was used to assign a score to each candidate. For this task, ROUGE was configured to report recall as a ratio of two integers: the number of word n-grams in a candidate that matched n-grams in the human-written model summaries (R), and the number of word n-grams in the human-written model summaries (F). BBN's Optimizer was used to determine the weights for the features that maximized

$$\frac{\sum_{r \in R} r}{\sum_{f \in F} f}.$$

The optimized weights support the interpretations of the penalties as used in the HMM Hedge decoder. There is a bias against compressions with a large sum of story word positions, which means that words near the start of the sentence are preferred. Small gaps are preferred over large gaps, and there is a bias in favor of compressions with fewer clumps. There is a strong bias in favor of sentences early in the documents. The negative weight for length in words combined with the positive weight for length in characters gives a preference to compressions containing large words rather than short words. Finally, the bigram, unigram and emit probabilities are log likelihoods. Very unlikely strings of headline words will have large negative bigram probabilities. The positive weights on the probabilities in the Headline and General language model show a bias in favor of more likely configurations of headline and general language words, and more likely headline words given the underlying story words.

3.4 Evaluation of HMM Hedge

This section will describe three automatic evaluations of HMM Hedge. The first evaluation shows that optimization of feature weights improves HMM Hedge performance over using the decoder parameters as the feature weights, but that using the learned feature weights as decoder parameters does not improve performance. The second evaluation explores how the filtering stage of MASC affects HMM Hedge performance. The third evaluation shows that using HMM Hedge to generate multiple compressions combined with feature-based selection outperforms using HMM Hedge to generate the single most likely compression.

3.4.1 Evaluation of HMM Hedge MASC with Optimized Feature Weights

An initial evaluation of HMM Hedge on the DUC2003 single-document test data shows that optimization for Rouge-1 recall of the feature weights on training data improves scores of Rouge-1 recall on testing data. However, using the optimized weights as parameters to HMM Hedge decoding does not improve Rouge-1 recall.

HMM Hedge was run on 624 documents from the DUC2003 single-document test set. Compression was performed on the first five sentences of each document. The initial values for the decoding parameters described in Section 3.2.1 are shown in Table 3.3. The small negative values of these parameters bias the decoder against large word positions, multi-word gaps and large number of clumps. The smaller

Parameter	Initial	Optimized
Word Position	log(0.95) = -0.05129	-0.04603
One Word Gap	log(0.99) = -0.01005	4.19282
Multi Word Gap	log(0.95) = -0.05129	1.20494
Clump	log(0.95) = -0.05129	-0.17963

Table 3.3: Decoding Parameters for HMM Hedge

Test Set	Initial Weights	Optimized Weights	Optimized Decoding Parameters
A	0.12276	0.25010	0.24513
B	0.13551	0.24621	0.23064
C	0.12520	0.25294	0.23253
D	0.13993	0.26394	0.24946
E	0.14564	0.25049	0.25739
Avg	0.13381	0.25274	0.24303

Table 3.4: Rouge-1 recall scores for 5-fold cross-validation

negative value for one-word gaps creates a weaker bias against one-word gaps.

HMM Hedge generated up to 55 compressions of each source sentence. All of the candidates were evaluated using Rouge-1. The features shown in Table 3.2 were tuned using 5-fold cross validation. The DUC2003 test set was divided into 5 subsets. Each subset was in turn withheld from the training data used to optimize the feature weights for Rouge-1. Thus a set of feature weights was learned for each subset based on the other four subsets. In addition a set of feature weights was trained on the entire corpus. These weights are shown in Table 3.2 as the optimized weights.

Headlines were selected from the candidates for each subset using two sets of feature weights: the initial feature weights shown in Table 3.2 and the optimized weights for the subset. The system output for each subset was evaluated with Rouge-1 recall. HMM Hedge was re-run on the documents using the optimized feature weights from Table 3.3 as the decoding parameters for the Viterbi decoder and as the feature weights for the summary selector. The results are shown in Table 3.4. The column Test Set shows which subset was withheld from training as test data, and the Rouge scores are for that set. The column Initial Weights and Optimized Weights shows the Rouge scores using the initial and optimized weights from Table 3.2 for selection. The column Optimized Decoding Parameters shows the Rouge scores using optimized feature weights as decoding parameters. The Avg row gives the average scores over the test sets. Using the optimized weights resulted in a significant improvement in Rouge-1 recall over using the initial weights. Using the optimized features as decoding parameters to the Viterbi decoder did not improve performance over using the initial decoding parameters. Developing a method to optimize the decoding parameters so that HMM Hedge produces the best set of candidates is an important task for future work.

Note that the optimization was performed using training data produced with the initial decoding parameters. When the optimized features are used as decoding parameters a different set of compressions is produced. The optimization that was used in this evaluation is sufficient to improve Rouge-1 scores on withheld testing data by using the optimized weights as feature weights. However the optimization is not sufficient to improve the performance of the HMM Hedge decoder when the weights are used as decoding parameters. This is an interesting area for future work.

3.4.2 Evaluation of Word Block Selection for HMM Hedge

Another area of interest in the configuration of HMM Hedge is selecting the block of document words on which to apply the HMM word-selection algorithm. The selection of a block of words corresponds to the filtering stage of the MASC framework. One approach is to select the first N words of the document. Another approach is to respect sentence boundaries by selecting the first N sentences of the document.

HMM Hedge was applied to the 624 documents of the DUC2003 test data, with the initial decoding parameters from Table 3.3 and candidate selection based on the features and weights given in Table 3.2. The systems differed in the filtering, i.e., the selection of the block of words to which HMM Hedge was applied. Five systems used blocks of initial words without respect for sentence boundaries and five systems used the initial sentences of the documents. A baseline consisting of the first 75 characters of each document was also evaluated. The system outputs were evaluated using Rouge-1.

The results of this evaluation are shown in Table 3.5. The evaluation shows that the best results are obtained by selecting a block of words that corresponds to the first sentence of the document. Of the systems that do not respect sentence boundaries, the 20-word block scores highest, and 20 words is approximately the average word length of sentences in English. This evaluation provides further evidence that the first document of a sentence is a good choice for lead sentence and that sentence compression give significantly higher performance than a baseline of

Filter Method	Rouge-1 Recall
First 75 characters	**0.21886** (0.20854-0.23001)
First 10 Words	**0.18837** (0.17833-0.19868)
First 20 Words	**0.25405** (0.24324-0.26609)
First 30 Words	**0.24864** (0.23867-0.25990)
First 40 Words	**0.24730** (0.23716-0.25736)
First 50 Words	**0.24354** (0.23396-0.25378)
First 60 Words	**0.24338** (0.23436-0.25340)
First 1 Sentence	**0.25503** (0.24480-0.26597)
First 2 Sentences	**0.19611** (0.18503-0.20715)
First 3 Sentences	**0.14378** (0.13400-0.15314)
First 4 Sentences	**0.12381** (0.11497-0.13234)
First 5 Sentences	**0.11455** (0.10575-0.12304)

Table 3.5: HMM Hedge using different filtering methods and a baseline of the first 75 characters evaluated on DUC2003 test data using Rouge-1 recall, with 95% confidence intervals.

selecting the first 75 characters.

3.4.3 Evaluation of single-candidate HMM Hedge and HMM Hedge MASC

A third evaluation confirms that HMM Hedge within the MASC framework performs better than HMM Hedge generating the single most likely compression. HMM Hedge was used to summarize 500 documents from the DUC2004 test data. Multi-candidate compression was performed on the first sentence of each document. A 75-character summary was selected from the candidates in three different ways. First, the candidate under the size limit with the highest Viterbi score was selected.

Selection Method	ROUGE-1 Recall
High Viterbi Score	**0.23090** (0.21620-0.24523)
HMM Features	**0.25181** (0.23651-0.26642)
HMM and URA Features	**0.26313** (0.24795-0.27780)

Table 3.6: HMM Hedge using different candidate selection methods evaluated on DUC2004 test data using ROUGE-1 recall, with 95% confidence intervals.

Second, the candidate under the size limit with the highest feature score based on the features in Table 3.3 was selected. Third, additional features dealing with query relevance, document centrality, and topic centrality were added to the feature set. These information retrieval features are calculated by the Uniform Retrieval Architecture (URA), and are described in more detail in Section 5.3.1. The system outputs of these three candidate selection methods were evaluated using ROUGE-1 recall. The results are shown in Table 3.6. This evaluation demonstrates the value of using a separate candidate selection stage in the MASC framework. The system which combines HMM-internal features with URA features for candidate selection performs significantly better than the system that simply uses the Viterbi score to determine the best compression.

This section has presented three evaluations of single-document summarization systems using HMM compression. These evaluations show that compression is useful process for a summarization system, with respect to ROUGE evaluation. The next section will consider how successful HMM summaries are from a human perspective.

3.5 Limitations of HMM Hedge

HMM Hedge generates compressions that are likely to occur based on a bigram model of a corpus of desirable compressions, such as newspaper headlines. This method often produces compressions that are readable and meaningful, but it does not guarantee that they will mean the same thing as the source. Consider the following examples from the DUC2003 test data.

(10) (i) The findings, some mental health experts say, lend support to the idea that doctors will eventually be able to offer preventive treatment to people who are judged to be at high risk for schizophrenia but have not yet fallen ill.

(ii) mental health experts say judge to fall ill

(11) (i) Health officials are closely monitoring diseases and have not reported any epidemics, the China Daily on Wednesday quoted Wang Zhao, the Ministry of Health official in charge of disease control, as saying.

(ii) Health officials reported epidemics China Daily on Health official says

Compression (10ii) gives the false impression that the story concerns a judge who is likely to become ill instead of preventive treatments for schizophrenia. Even worse, Compression (11ii) gives exactly the wrong information: that Chinese health officials reported epidemics, when according to the source they did not report epidemics.

The root of these errors lies in HMM Hedge's failure to take advantage of syntactic information from the source sentences that might indicate that judge is not a noun in Sentence (10i) or that removing "RB not" from the construction [VP [AUX [RB not] VP]] without also removing the entire parent VP will invert the meaning.

In Chapter 4 a headline generation system will be presented that is capable of avoiding errors of this kind.

3.6 Summary

In this chapter the underlying select-words-in-order approach to sentence compression was described along with feasibility studies that show humans can create fluent and informative summaries using this method. HMM Hedge, a statistical method for selecting words in order from the story was presented along with the decoding parameters and morphological variation mechanism that allow it to mimic Headlinese.

It was shown that HMM Hedge performs better on blocks of words that respect sentence boundaries, and that the best results are achieved by selecting the first sentence of a document during the MASC filtering stage. The performance of HMM Hedge on Rouge was improved by more than a factor of 2 by using a 5-fold cross validation to optimize the parameters involved in selecting among the 55 candidates available for each sentence.

An automatic evaluation of HMM Hedge along with other single document summarization systems appears in Section 4.6.3. A human extrinsic evaluation of HMM Hedge appears in Section 4.7. Section 5.4.1 will discuss the extension of HMM Hedge to the context of multi-document summarization.

Chapter 4

Trimmer Sentence Compression and Topiary Candidate Generation

This chapter presents Trimmer, a parse-and-trim approach to sentence compression, and Topiary, an extension to Trimmer that combines sentence compressions with topic terms. Trimmer and Topiary are used to generate compressed candidates within the MASC framework. The first section describes the linguistic observations of newspaper headlines that motivated the design of Trimmer, shown in Figure 4.1. This is followed by a detailed description of the Trimmer rules that perform syntactic trimming and the Trimmer algorithm according to which the rules are applied. Topiary, shown in Figure 4.2, is presented along with Unsupervised Topic Detection (UTD) and OnTopic, the supporting technologies that respectively discover topic terms from a corpus and assign topic terms to documents. UTD and OnTopic were developed by BBN. Trimmer and Topiary are incorporated into the MASC framework by enabling them to present multiple candidates to the sentence selector. The chapter concludes with automatic and human evaluations of Trimmer, Topiary, and HMM Hedge on a single-document summarization task.

The general approach used by Trimmer for the generation of a summary from a single document is to produce a *headline* by selecting words in order from the text of the story. Trimmer uses a linguistically-motivated algorithm to remove grammatical constituents from the lead sentence until a length threshold is met. Topiary is a

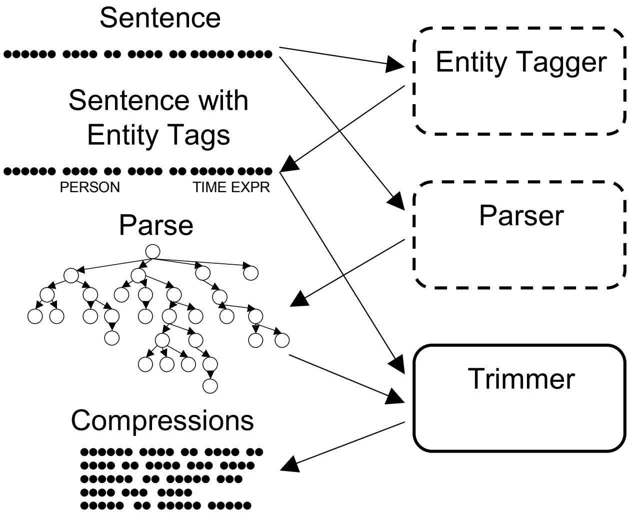


Figure 4.1: Trimmer architecture

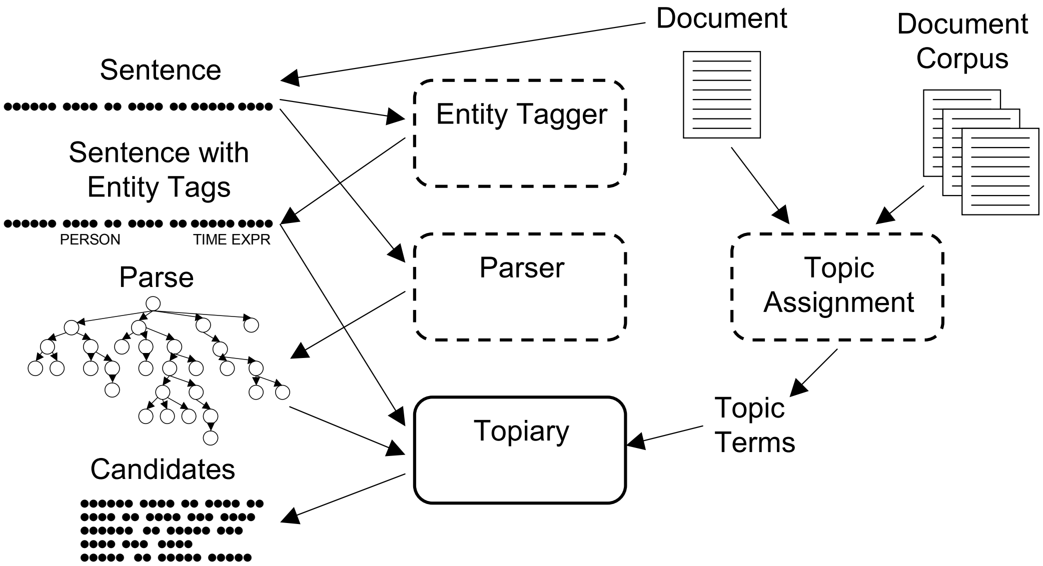


Figure 4.2: Topiary architecture

variant of Trimmer that combines fluent text from a compressed sentence with topic terms to produce headlines.

Trimmer performs sentence compression by iteratively removing grammatical constituents from the parse tree of a sentence using linguistically-motivated rules until a length threshold has been met. When applied to the lead sentence, or first non-trivial sentence of a story, the Trimmer algorithm generates a very short summary, or headline. Trimmer can leverage the output of any constituency parser that uses the Penn Treebank conventions. At present, Charniak's parser (Charniak, 2000) is used. Trimmer and Topiary will first be presented in forms that produce a single compression of a sentence, followed by more flexible versions that generate multiple compressions. Implementation-level descriptions of the Trimmer sentence compression tool and the Topiary candidate generation tool are given in Appendix Band Appendix C.

4.1 Linguistic Motivation for Trimmer

The insights that form the basis and justification for the Trimmer rules come from a study that compared the relative prevalence of certain constructions in human-written summaries and lead sentences in stories. This study used 218 human-written summaries of 73 documents from the TIPSTER corpus (Harman and Liberman, 1993) dated January 1, 1989. The 218 summaries and the lead sentences of the 73 stories were parsed using the BBN SIFT parser (Miller et al., 2000). The parser produced 957 noun phrases (NP nodes in the parse trees) and 315 clauses

Level	Phenomenon	Summary		Lead Sentence	
Sentence	preposed adjuncts	0/218	0%	2/73	2.7%
	conjoined S	1/218	0.5%	3/73	4%
	conjoined VP	7/218	3%	20/73	27%
Clause	temporal expression	5/315	1.5%	77/316	24%
	trailing PP	165/315	52%	184/316	58%
	trailing SBAR	24/315	8%	49/316	16%
Noun Phrase	relative clause	3/957	0.3%	29/817	3.5%
	determiner	31/957	3%	205/817	25%

Table 4.1: Counts and prevalence of phenomena found in summaries and lead sentences.

(S nodes in the parse trees) for the 218 summaries. For the 73 lead sentences, the parser produced 817 noun phrases and 316 clauses.

At each level (sentence, clause, and noun phrase), different types of linguistic phenomena were counted.

At the sentence level, the numbers of preposed adjuncts, conjoined clauses, and conjoined verb phrases were counted. Children of the root S node that occur to the left of the first NP are considered to be preposed adjuncts. The bracketed phase in “[According to police] the crime rate has gone down” is a prototypical example of a preposed adjunct.

At the clause level, temporal expressions, trailing SBAR nodes, and trailing PP nodes were counted. Trailing constituents are those not designated as an argument of a verb phrase. At both the sentence and clause levels, conjoined S nodes and conjoined VP nodes were counted.

At the NP level, determiners and relative clauses were counted.

The counts and prevalence of the phenomena in the human-generated headlines and lead sentences are shown in Table 4.1. The results of this analysis illuminated the opportunities for trimming constituents, and guided the development of the Trimmer rules, detailed in Section 4.2.

4.2 Trimmer Algorithm

Trimmer applies syntactic compression rules (Trimmer rules) to a parse tree to delete or *mask* constituents from a parse tree. The Trimmer rules mask constituents from general English sentences to make them appear more like the newspaper Headlines in style and content. When a leaf node in a tree is marked as masked, it will not appear as part of the surface representation of that parse tree. Two masking operations are implemented. The first, called *mask*, marks a node and all of its descendants as masked. The second, called *mask outside*, masks all the nodes in the tree except for a node and its descendants. In other words it masks all the nodes in the tree that are outside of its argument and its argument's descendants.[1] Examples of a mask operation and a mask outside operation are shown in Figures 4.3 and 4.4.[2]

The Trimmer sentence compression tool applies Trimmer rules to a parse tree according the following algorithm:

1. Remove temporal expressions
2. Select Root S node

[1] The operation is actually more powerful in that it takes two arguments and masks all the descendants of its first argument except for the descendants of the second argument. However in the actual calls to this operation the higher node is always the root of the tree, so it just masks outside of its second argument.

[2] The tree images are the output of a diagnostic web tool developed by the author for use in debugging Trimmer. More trees are available to examine at `http://www.umiacs.umd.edu/dmzajic/trees/trees.html`

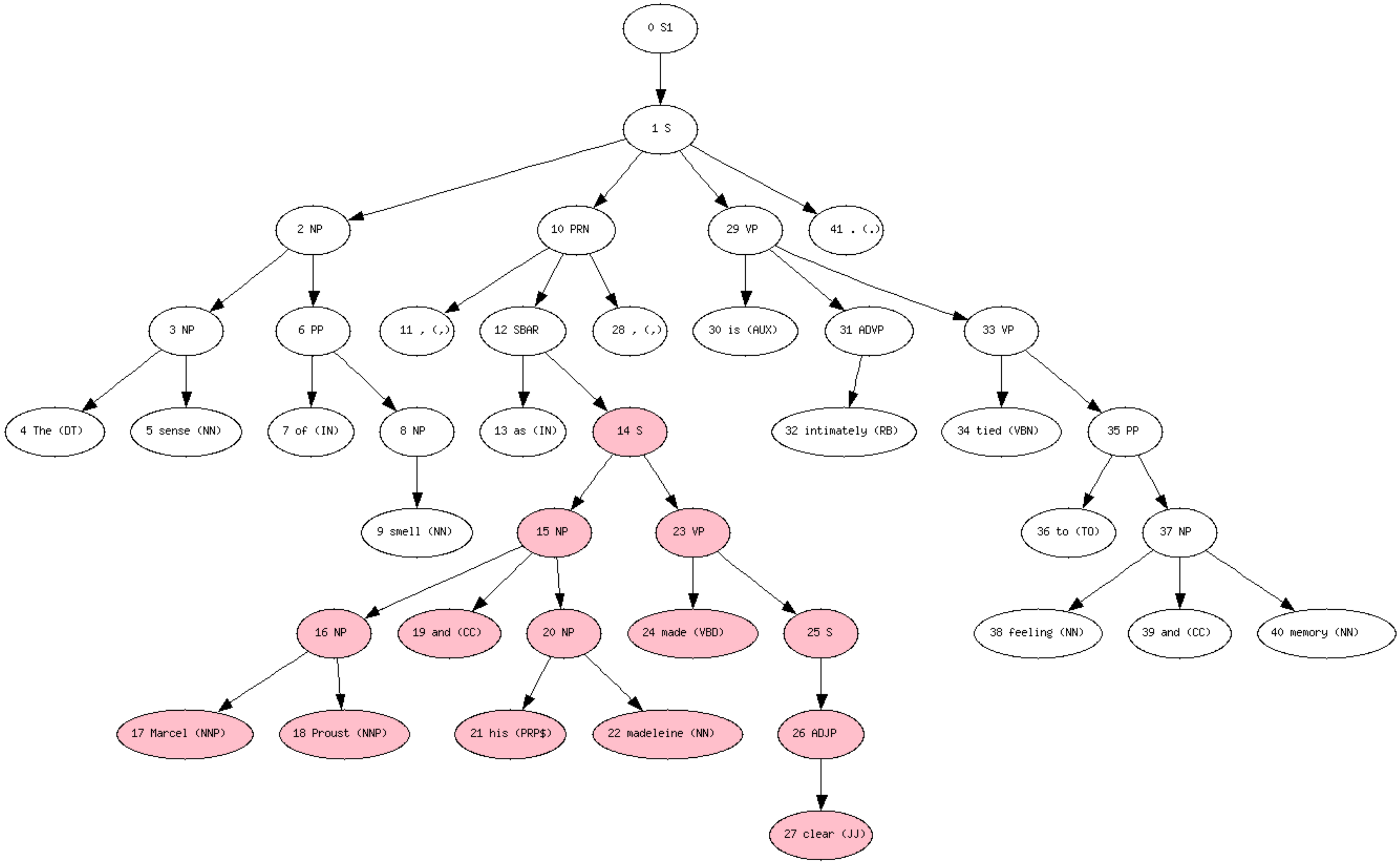


Figure 4.3: A Parse tree with a mask operation applied to it. The masked nodes are shown in gray. Node 14 and all its descendants have been trimmed from the tree.

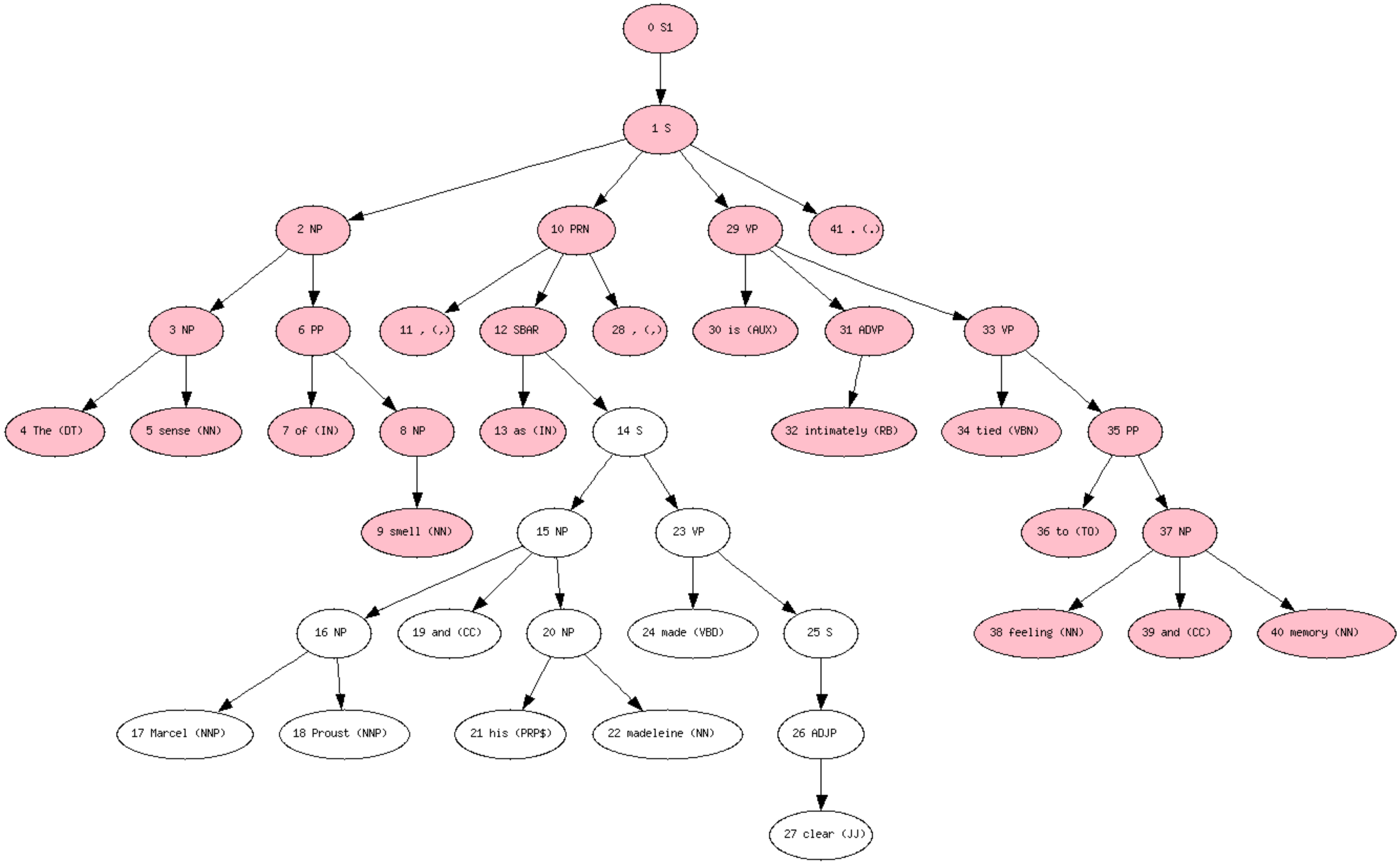


Figure 4.4: A Parse tree with a mask outside operation applied to it. All nodes except node 14 and its descendants have been trimmed from the tree.

3. Remove preposed adjuncts
4. Remove some determiners
5. Remove conjunctions
6. Remove modal verbs
7. Remove complementizer *that*
8. Apply the XP over XP rule
9. Remove PPs that do not contain named entities
10. Remove all PPs under SBARs
11. Remove SBARSs
12. Backtrack to state before Step 9
13. Remove SBARs
14. Remove PPs that do not contain named entities
15. Remove all PPs

Steps 1 and 4 of the algorithm remove low-content units from the parse tree. Temporal expressions—although certainly not content-free—are not usually vital for summarizing the content of an article. Since the goal is to provide an informative headline, the identification and elimination of temporal expressions (Step 1) allows other more important details to remain in the length-constrained headline. The use of BBN's IdentiFinder (Bikel et al., 1999) for removal of temporal expressions is described in Section 4.2.2.

The determiner rule (Step 4) removes leaf nodes that are assigned the part-of-speech tag DT and have the surface form *the*, *a* or *an*. The intuition for this rule is that the information carried by articles is expendable in summaries, even though this makes the summaries ungrammatical for general English. Omitting articles is one of the most salient features of newspaper headlines. Sentences (12) and (13), taken from the New York Times website on September 27, 2006, illustrate this phenomenon. The italicized articles did not occur in the actual newspaper headlines.

(12) *The* Gotti Case Ends With *a* Mistrial for *the* Third Time in a Year

(13) *A* Texas Case Involving Marital Counseling Is *the* Latest to Test *the* Line Between Church and State

Step 2 identifies nodes in a parse tree of a sentence that could serve as the root of a compression for the sentence. Such nodes will be referred to as Root S nodes. When a Root S node is chosen all the nodes except the Root S node and its descendants are trimmed. This is the only rule that makes use of the mask outside operation. A node in a tree is a Root S node if it is labeled S in the parse tree and has children labeled NP and VP, in left-to-right order.[3] The human-generated headlines that were studied always conform to this rule. It has been adopted as a constraint in the Trimmer algorithm that the lowest leftmost Root S node is taken to be the root node of the headline. An example of this rule application is shown in (14). The boldfaced material in the parse is retained and the italicized material is eliminated.

(14) (i) Input: Rebels agreed to talks with government officials, international observers said Tuesday.

(ii) Parse: *[S* **[S [NP Rebels][VP agreed to talks with government officials]]***, international observers said Tuesday.]*

(iii) Output: Rebels agreed to talks with government officials.

[3] This requirement is compatible with the *Projection Principle* in Linguistic theory (Chomsky, 1981): predicates project a subject (directly dominated by S) in the surface structure.

When the parser produces a correct parse tree, this rule selects a valid starting point for compression. However, the parser sometimes produces incorrect output, as in the cases below (from the DUC2003 test data):

(15) (i) Parse: **[S[SBAR What started as a local controversy][VP has evolved into an international scandal.]]**

(ii) Parse: **[NP[NP Bangladesh][CC and][NP[NP India][VP signed a water sharing accord.]]]**

In (15i), an S exists, but it does not conform to the requirements of the Root S rule because it does not have as children an NP followed by a VP. The problem is resolved by selecting the lowest leftmost S, ignoring the constraints on the children. In (15ii), no S is present in the parse. This problem is resolved by selecting the root of the entire parse tree as the root of the headline. These parsing errors occur infrequently—only 6% of the sentences in the DUC2003 evaluation data exhibit these problems, based on parses generated by the BBN SIFT parser.

The motivation for removing preposed adjuncts (Step 3) is that all of the human-generated headlines omit the *preamble* of the sentence. Preposed adjuncts are constituents that precede the first NP (the subject) under the Root S chosen in Step 2; the preamble of a sentence consists of its preposed adjuncts. The impact of preposed adjunct removal can be seen in example (16).

(16) (i) Input: According to a now finalized blueprint described by U.S. officials and other sources, the Bush administration plans to take complete, unilateral control of a post-Saddam Hussein Iraq.

(ii) Parse: **[S***[PP According to a now finalized blueprint described by U.S. officials and other sources], [Det the]* **Bush administration plans to take complete, unilateral control of***[Det a]* **post-Saddam Hussein Iraq.]**

(iii) Output: Bush administration plans to take complete unilateral control of post-Saddam Hussein Iraq.

The remaining steps of the algorithm remove linguistically peripheral material through successive deletions of constituents until the sentence is shorter than a length threshold. Each stage of the algorithm corresponds to the application of one of the rules. Trimmer first finds the pool of nodes in the parse to which a rule can be applied. The rule is then iteratively applied to the deepest, rightmost remaining node in the pool until the length threshold is reached or the pool is exhausted. After a rule has been applied at all possible nodes in the parse tree, the algorithm moves to the next step.

In the case of a conjunction with two children (Step 5), one of the children will be removed. If the conjunction is *and*, the second child is removed. If the conjunction is *but*, the first child is removed. This rule is illustrated by the following examples, where the struck out text is trimmed.

(17) When Sotheby's sold a Harry S Truman signature that turned out to be a reproduction, the prestigious auction house apologized *and bought it back*.

(18) President Clinton *expressed sympathy after a car-bomb explosion in a Jerusalem market wounded 24 people but* said the attack should not derail the recent land-for-security deal between Israel and the Palestinians.

The modal verb rule (Step 6) applies to verb phrases in which the head is a modal verb, and the head of the child verb phrase is a form of *have* or *be*. In such cases, the modal and auxiliary verbs are removed. Sentences (19) and (20) show examples of this rule application. Note that although in Sentence (20), the omission of trimmed material changes the meaning, given a tight space constraint, the loss of the modality is preferable to the loss of other content information.

(19) People's palms and fingerprints *may be* used to diagnose schizophrenia.

(20) Agents *may have* fired potentially flammable tear gas cannisters.

The complementizer rule (Step 7) removes the word *that* when it occurs as a complementizer. Sentence (21) shows an example in which two complementizers can be removed.

(21) Hoffman stressed *that* the study is only preliminary and can't prove *that* the treatment would be useful.[4]

The XP-over-XP rule (Step 8) is a linguistic generalization that allows a single rule to cover two different phenomena. XP in the name of the rule is a variable that can take two values: NP and VP. In constructions of the form [XP [XP ...] ...], the other children of the higher XP are removed. Note that the child XP must be the first child of the parent XP. When XP = NP the rule removes relative clauses (as in Sentence (22)) and appositives (as in Sentence (23)).

(22) Schizophrenia patients *whose medication couldn't stop the imaginary voices in their heads* gained some relief.

(23) A team led by Dr. Linda Brzustowicz*, assistant professor of neuroscience at Rutgers University's Center for Molecular and Behavioral Neuroscience in Newark,* studied DNA of dozens of members of 22 families.

The rules that remove prepositional phrases and subordinate clauses (Steps 9 through 15) are sometimes prone to removing important content. Thus, these rules are applied last, only when there are no other types of rules to apply. Moreover, these rules are applied with a backoff option to avoid over-trimming the parse tree.

[4] For clarity, this example is presented as if the complementizer rule were applied in isolation to the source sentence. In the context of actual Trimmer processing several rules have already been applied to the sentence before reaching the complementizer rule, and the output in context of the complementizer rule is: "Hoffmann stressed study is only preliminary and can't prove treatment useful."

First, the PP rule is applied (Steps 9 and 10),[5] followed by the SBAR rule (Step 11). If the desired sentence length has not been reached, the system reverts to the parse tree as it was before any prepositional phrases were removed (Step 12) and applies the SBAR rule (Step 13). If the desired length still has not been reached, the PP rule is applied again (Steps 14 and 15).

The intuition behind this ordering is that, when removing constituents from a parse tree, it is preferable to remove smaller fragments before larger ones, and prepositional phrases tend to be smaller than subordinate clauses. Thus, Trimmer first attempts to achieve the desired length by removing smaller constituents (PPs), but if this cannot be accomplished, the system restores the smaller constituents, removes a larger constituent, and then resumes the deletion of the smaller constituents. To reduce the risk of removing prepositional phrases that contain important information, BBN's IdentiFinder is used to distinguish PPs containing temporal expressions and named entities, as described in Section 4.2.2.

4.2.1 Head Nodes

Under certain circumstances, rules are not allowed to fire, such as when doing so would compress the sentence to an empty string. A more interesting constraint makes use of *head nodes*. Each non-leaf node in a parse tree has exactly one head node. The head node is determined using a configurable set of head rules. A head rule for a node label, such as S or NP, specifies sets of labels that can serve as the head, and whether the head is the rightmost or leftmost of the children with that

[5] The reason for breaking PP removal into two stages is discussed in Section 4.2.2.

label. The head is determined by the first set of labels for which there is a matching child node. The head rules used by Trimmer are shown in Figure 4.5. For example, if a node has label PP, then the leftmost of its children labeled IN, RP or TO is the head. If there are no children with those labels, then the rightmost child labeled PP is the head.

Head nodes are used to constrain the XP over XP rule (Step 8). The XP over XP with XP = NP rule applies only in cases where the child NP is the head node of the parent NP. This ensures that the rule will not mask a head node while leaving its siblings unmasked. For example "The city's department of consumer affairs," is parsed by Charniak's parser as [NP [NP [NP The city's] Department] [PP of consumer affairs]], with *Department* as the head of the second NP. If the NP-Over-NP rule applied here, the text would appear as "The city's of consumer affairs." However the head node constraint prevents this and the head of the NP, *Department* is preserved.

4.2.2 Use of Temporal Expression and Named Entity Tagging in Trimmer

Named entity tagging is used both for the elimination of temporal expressions and for conservative deletion of PPs containing named entities. At present BBN's IdentiFinder is used for the tagging of temporal expressions and named entities. The elimination of temporal expressions (Step 1) is a two-step process: (a) use IdentiFinder to mark temporal expressions; and (b) remove [PP ... [NP [X] ...] ...]

- *default* (r AUX AUXG BES CC CD DT EX FW IN JJ JJR JJS LS MD NN NNS NNP NNPS PDT POS PRP PRP$ RB RBR RBS RP SYM TO UH VB VBD VBG VBN VBP VBZ WDT WP WP$ WRB # $) (r ADJP ADVP CONJP FRAG INTJ LST NAC NP NX PRN PRT QP RRC S S1 SBAR SBARQ SINV SQ UCP VP WHADJP WHADVP WHNP WHPP X) (r PP) (r . , : -RRB- -LRB- “ ” XX HVS GW) (r)
- ADJP (r JJ JJR JJS) (r ADJP) (r RB VBN)
- ADVP (r RB RBB) (r ADVP)
- CONJP (r CONJP)
- FRAG (r FRAG)
- INTJ (r INTJ)
- LST (r LS) (r LST)
- NAC (r NN NNP NNPS NNS PRP) (r NAC) (r ADJP CD FW JJ NP)
- NP (r $ NN NNP NNPS NNS POS PRP) (r NP) (r ADJP CD JJ NX)
- NX (r NN NNP NNPS NNS PRP) (r NX) (r ADJP CD FW JJ NP)
- PP (l IN RP TO) (r PP)
- PRN (r PRN)
- PRT (r RP) (r PRT) (r IN RB)
- QP (r QP) (r $ NN)
- RRC (r RRC)
- S (r VP) (r S) (r SBARQ SINV X)
- SBAR (r IN WHNP) (r SBAR) (r WHADJP WHADVP WHPP)
- SBARQ (r SQ VP) (r SBARQ) (r S SINV X)
- SINV (r VP) (r SINV) (r SBAR)
- SQ (r AUX BES HVS MD) (r SQ) (r VP)
- UCP (r UCP)
- VP (r AUX AUXG BES HVS MD TO VB VBD VBG VBN VBP VBZ) (r VP)
- WHADJP (r WRB) (r WHADJP)
- WHADVP (r WRB) (r WHADVP)
- WHNP (r WDT WP WP$) (r WHNP)
- WHPP (r IN TO) (r WHPP)
- X (r X)

Figure 4.5: Head rules used by Trimmer.

and [NP [X]] where X is tagged as part of a temporal expression. The following examples illustrate the application of temporal expression removal rule:

(24) (i) Input: The State Department on Friday lifted the ban it had imposed on foreign fliers.

(ii) Parse: **[S [NP***[Det The]* **State Department** *[PP [IN on] [NP [NNP Friday]]]* **[VP lifted** *[Det the]* **ban it had imposed on foreign fliers.**]]

(iii) Output: State Department lifted ban it had imposed on foreign fliers.

(25) (i) Input: An international relief agency announced Wednesday that it is withdrawing from North Korea.

(ii) Parse: **[S [NP** *[Det An]* **international relief agency**]**[VP announced** *[NP [NNP Wednesday]]* **that it is withdrawing from North Korea.**]]

(iii) Output: International relief agency announced that it is withdrawing from North Korea.

Named entity tagging is also used to ensure that prepositional phrases containing named entities are not removed during the first round of PP removal (Step 9). However, prepositional phrases containing named entities that are descendants of SBARs are removed before the parent SBAR is removed, since it is safer to remove a smaller constituent before removing a larger constituent that subsumes it. Sentence (26) shows an example of a SBAR subsuming two PPs, one of which contains a named entity.

(26) The commercial fishing restrictions in Washington will not be lifted [SBAR unless the salmon population increases [PP to a sustainable number] [PP in the Columbia River]].

If the PP rule were not sensitive to named entities, the PP *in the Columbia River* would be the first prepositional phrase to be removed, because it is the lowest rightmost PP in the parse. However, this PP provides an important piece of information: the location of the salmon population. The rule in Step 9 will skip the last prepositional phrase and remove the penultimate PP *to a sustainable number*.

This concludes the description of the Trimmer rules and Trimmer's syntactic sentence compression algorithm. Given a length limit, the system will produce a single compressed version of the target sentence.

4.3 Topiary

The Trimmer approach to compression is used as a component in another variant of single-document summarization called Topiary. This system combines Trimmer with a topic discovery approach (described in Section 4.3.1) to produce a fluent summary along with additional context.

The Trimmer algorithm is constrained to build a headline from a single sentence. However, it is often the case that no single sentence contains all the important information in a story. Relevant information can be spread over multiple sentences, linked by anaphora or ellipsis. In addition, the choice of lead sentence may not be ideal and the trimming rules are imperfect.

On the other hand, approaches that construct headlines from lists of topic terms (Lewis, 1999; Schwartz et al., 1997) also have limitations. For example, Unsupervised Topic Discovery (UTD), described in Section 4.3.1, rarely generates any topic terms that are verbs. Thus, topic lists are good at indicating the general subject but rarely give any direct indication of what events took place. Intuitively, a summary needs both fluent text to tell what happened and topic terms to provide context.

4.3.1 Topic Term Generation: UTD and OnTopic

OnTopic (Schwartz et al., 1997) uses an HMM to assign topics to a document; topic models are derived from an annotated corpus. However, it is often difficult to acquire such data, especially for a new genre or language. UTD (Sista et al., 2002) was developed to overcome this limitation: it takes as input a large unannotated corpus and automatically creates a set of topic models with meaningful names.

The UTD algorithm has several stages. First, it analyzes the corpus to find multi-word sequences that can be treated as single tokens. It does this using two methods. One method is a minimum description length criterion, which detects phrases that occur frequently relative to the individual words. The second method uses BBN's IdentiFinder to detect multi-word names. These names are added to the text as additional tokens. They are also likely to be chosen as potential topic names. In the second stage, UTD finds those terms (both single-word and multi-word) with high *tf.idf*. Only those topic names that occur as high-content terms in at least four different documents are kept. The third stage trains topic models corresponding to these topic terms. The modified Expectation Maximization procedure of BBN's OnTopic system is used to determine which words in the documents often signify these topic names. This produces topic models. Fourth, these topic models are used to find the most likely topics for each document, which is equivalent to assigning the name of the topic model to the document as a topic term. This often assigns topics to documents where the topic name does not occur in the document text.

It has been shown by Sista et al. (2002) that the topic names derived by this procedure were usually meaningful and that the topic assignment was about as good as when the topics were derived from a corpus that was annotated by people. They have also used this procedure on different languages and shown the same behavior. Since UTD is unsupervised, it can run equally well on a new language, as long as the documents can be divided into strings that approximate words.

The topic list in (27) was generated by UTD and OnTopic for a story about the FBI investigation of the 1998 bombing of the U.S. embassy in Nairobi. Each topic term is associated with a score, with higher scores indicating more apt topic terms.

(27) BIN LADEN (0.630) EMBASSY (0.596) BOMBING (0.472) POLICE (0.138) OFFICIALS (0.084) PRISON (0.059) HOUSE (0.052) FIRE (0.029) KABILA (0.026)

Topiary uses UTD to generate topic terms for the collection of documents to be summarized, and OnTopic to assign the topic terms to the documents. The next section will describe how topic terms and sentence compressions are combined to form Topiary summaries.

UTD can also be considered a summarization system in that the lists of topic terms it discovers and assigns to a document can be concatenated together to form a summary. A system in which UTD topic terms are sorted in order of their scores and truncated to 75 characters is evaluated in Section 4.6.3.

4.3.2 Topiary Algorithm

As each Trimmer rule is applied to a sentence, the resulting state of the sentence is stored as a compressed variant of the source sentence. Topiary selects from the variants the longest one such that there is room to prepend the N highest scoring non-redundant topic terms, where N is a parameter to the system. Suppose $N = 1$, the highest scoring topic term is “terrorism” and the length threshold is 75 characters. To make room for the topic “terrorism,” the length threshold is lowered by 10 characters: 9 characters for the topic and 1 character as a separator. Thus, Topiary seeks the longest trimmed variant under 65 characters that does not contain the word “terrorism.” If there is no such candidate, i.e., all the trimmed variants contain the word terrorism, Topiary would consider the second highest scoring topic word, “bomb.” Topiary would seek the longest trimmed variant under 70 characters that does not contain the word “bomb.” After Topiary has selected a trimmed variant and prepended a topic to it, it checks to see how much unused space remains under the threshold. Additional topic words are provided between the first topic word and the compressed sentence until all space is exhausted.

This process results in a headline that usually contains one or more main topics about the story and a short sentence that says what happened concerning them.[6] The combination is often more concise than a fully fluent sentence and compensates for the fact that the information content from the topic and the compressed sentence do not occur together in any single sentence from the source text.

As examples, sentences (28) and (29) are the outputs of Trimmer and Topiary, respectively, for the same story in which UTD selected the topic terms in (27).

(28) FBI agents this week began questioning relatives of the victims

[6] It is possible that for a given sentence the Trimmer rules cannot compress it to the desired length, in which case no topic is added. An alternate version of Topiary which guaranteed at least N topic terms did not perform significantly differently from the basic version.

(29) BIN LADEN, EMBASSY, BOMBING: FBI agents this week began questioning relatives

By combining topics and parse-and-trim compression, Topiary achieved the highest score on the single-document summarization task (i.e., headline generation task) in DUC2004 (Zajic et al., 2004). The Topiary concept is not exclusively linked to UTD as the source of topic terms. Recently Wang et al. (2005) have explored other topic term sources in combination with Topiary-style summarization. An interesting future direction for Topiary development would be the promotion of relevant content from within a constituent to serve as a topic term. For example, consider Sentences (30i) and (30ii).

(30) (i) Schizophrenia patients whose medication couldn't stop the imaginary voices in their heads gained some relief after researchers repeatedly sent a magnetic field into a small area of their brains.

(ii) Magnetic Field: Schizophrenia patients gained some relief.

In Sentence (30ii) "magnetic field" has been promoted from within the prepositional phrase "after researchers repeatedly sent a magnetic field into a small area of their brains" to the status of a topic term, and the prepositional phrase has been trimmed. Such "promotion from within" could preserve an important concept while trimming a lengthy constituent.

4.3.3 Balance of Topics and Compressed Text

The Topiary algorithm takes the number of requested topic terms as a parameter. Trimmer is constrained to remove constituents according to its rules and cannot guarantee a compression of exactly the right length in characters, so some-

N	Topic Count	Topic Character Count	ROUGE-1 Recall
0	1.552	12.786	0.24533
1	2.172	18.298	0.25033
2	2.680	22.828	0.24841
3	3.066	26.392	0.24347
4	3.282	28.298	0.24057
5	3.370	28.998	0.23812

Table 4.2: Evaluation using ROUGE-1 recall of Topiary on the DUC2004 test data. The systems vary on how many topics were requested. The average number of topic terms and average size in characters of the topic term component are also shown.

times a Topiary summary will have fewer or more topic terms than the requested value. The best value of N was determined by running Topiary on 500 news stories from the DUC2004 evaluation with N varying from 0 to 5 and a length threshold of 75 characters. The results are shown in Table 4.2.

The topic counts are the average number of topic terms in the Topiary output and the topic character counts are the number of characters (out of 75) taken up by topic terms. Note that Topiary usually includes more topic terms than requested. The differences among the systems according to ROUGE-1 recall are not significant, however this experiment supports the initial design decision that the best number of topic terms to request is 1 and the best number of topic terms to get is 2.

4.4 Multiple Alternative Compressions with Trimmer and Topiary

Trimmer and Topiary have so far been described as producing a single compressed version of a source sentence. Single document summaries are produced by generating the Trimmer or Topiary compression of the lead sentence. Within the MASC framework, the sentence filtering stage is lead sentence selection, and the sentence compression stage is Trimmer or Topiary. Since a single compression is generated, the sentence selection stage is trivial. The MASC framework is fully realized for single-document summarization when Trimmer and Topiary are used to generate multiple compressed candidates from the lead sentence, and a linear combination of features is used to select among the candidates. Sections 4.4.1 and 4.4.2 will discuss how Trimmer and Topiary are used to generate multiple candidate compressions of a source sentence. Section 4.5 will then deal with the problem of selecting among the candidates.

4.4.1 Multiple Alternative Compressions with Trimmer

So far, parse-and-trim systems have been shown using Trimmer rules in a specific order, trimming until a length threshold has been reached. With Topiary, the length threshold is adjusted so that space is available for the inclusion of a fixed number of topic terms. Under both systems, given a length threshold (and, for Topiary, the required number of topic terms), a single candidate is produced. Although single-candidate Topiary with length threshold of 75 characters and 1 required topic term was shown to be the best performer at DUC2004, the best compression could not be guaranteed because no single rule ordering could provide the best compression for all sentences. In addition, this single-candidate version of Topiary always included a minimum of one topic term in every summary.

The simplest way to use Trimmer rules to produce multiple compressed candidates is to treat the result of each Trimmer rule application as a candidate. This method for producing multiple candidates is called intermediate stage Trimmer. Intermediate stage Trimmer has the limitation that rules are still applied in a fixed order. Once a constituent has been trimmed, it is gone from all subsequent candidates. A more flexible approach is to modify some of the Trimmer rules so that they produce multiple candidates to which other rules can be applied.

Of the 14 linguistically-motivated Trimmer rules, three have been reconfigured as multi-candidate rules:

- **Root-S:** Identifies multiple parse-tree nodes that may serve as a (new) top node of compressed sentence

- **Preamble:** Identifies pre-sentential modifiers that may be removed or retained

- **Conjunction:** Identifies conjuncts that may either be removed or retained

Examples of each case are shown in Figure 4.6. For each of these rules, the single-candidate version (where only one option is considered) or the multi-candidate version (where all options are considered) can be used. An equivalent way of expressing this is to say that Trimmer can independently enable or disable each of the three multi-candidate rules.

Enabling/disabling these three rules results in eight possible configurations of the Trimmer approach: No-MC-Rules (intermediate stage Trimmer), Root-S, Preamble, Conjunction, Root-S/Preamble, Root-S/Conjunction, Preamble/Conjunction, and All-MC-Rules. The candidates resulting from these eight configurations serve

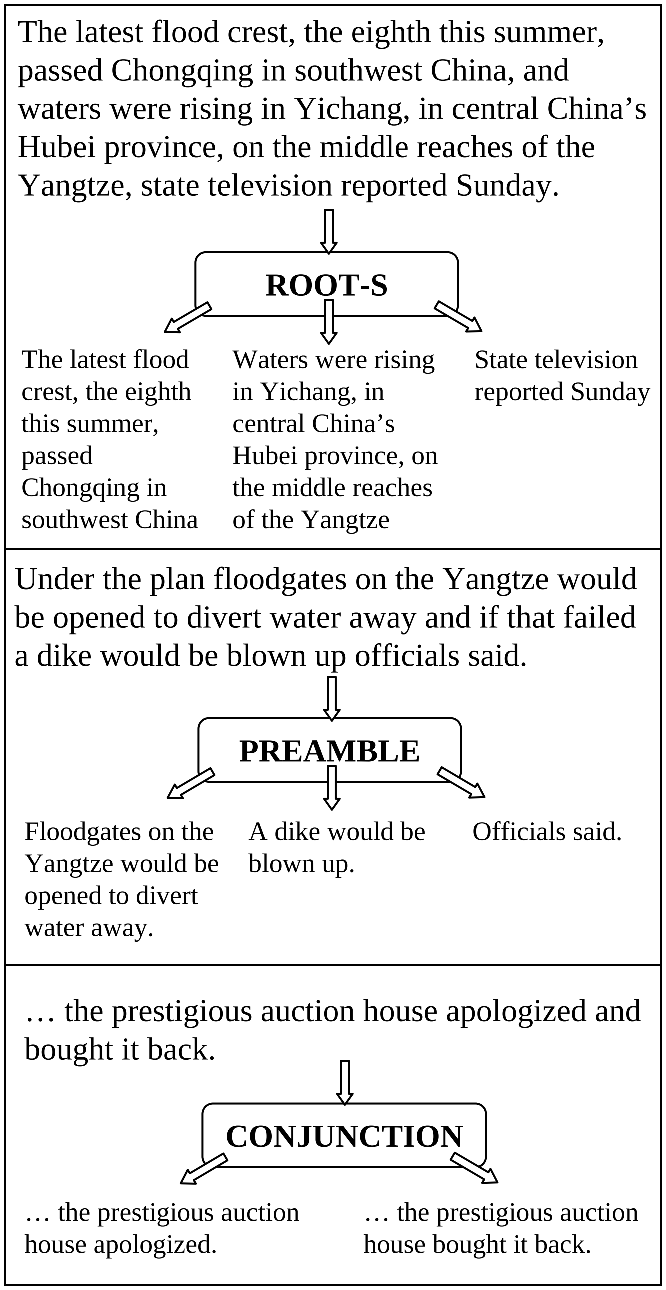

The latest flood crest, the eighth this summer, passed Chongqing in southwest China, and waters were rising in Yichang, in central China's Hubei province, on the middle reaches of the Yangtze, state television reported Sunday.
ROOT-S
The latest flood crest, the eighth this summer, passed Chongqing in southwest China
Waters were rising in Yichang, in central China's Hubei province, on the middle reaches of the Yangtze
State television reported Sunday
Under the plan floodgates on the Yangtze would be opened to divert water away and if that failed a dike would be blown up officials said.
PREAMBLE
Floodgates on the Yangtze would be opened to divert water away.
A dike would be blown up.
Officials said.
… the prestigious auction house apologized and bought it back.
CONJUNCTION
… the prestigious auction house apologized.
… the prestigious auction house bought it back.

Figure 4.6: Three Examples of Multi-Candidate Rules: Root-S, Preamble, and Conjunction

1. TIMEX Rule
2. Determiner Rule
3. Possessive Pronoun Rule
4. Modal Be/Have Rule
5. Auxiliary Verb Rule
6. Complementizer Rule
7. XP Over XP Rule
8. PP Rule with Named Entity Protection
9. PP Rule without Named Entity Protection
10. SBAR Rule

Figure 4.7: Rule ordering applied to starting-point trees in multi-candidate Trimmer sentence compression.

as starting point trees to which a fixed-order algorithm of the remaining rules is applied. The application order of the multi-candidate Trimmer rules does not matter in the generation of starting point trees; however, the subsequent rules fire according to a deterministic algorithm, shown in Figure 4.7.

The number of starting point trees depends on the number of places in the parse tree to which the multi-candidate rules can be applied. For example, if there are 4 Root S nodes, 2 of which have preambles, and there are 3 conjunctions in the parse, then there will be $4 \times 2^2 \times 3^3 = 432$ starting point trees. The actual number of starting point trees per sentence is generally much lower. In the first 5 sentences of the documents of the DUC2003 test data, there were on average 2.58 Root S nodes, 0.39 preambles, and 0.10 conjunctions per sentence, with an average number of 2.83 starting point trees per sentence. Figure 4.8 shows the distribution

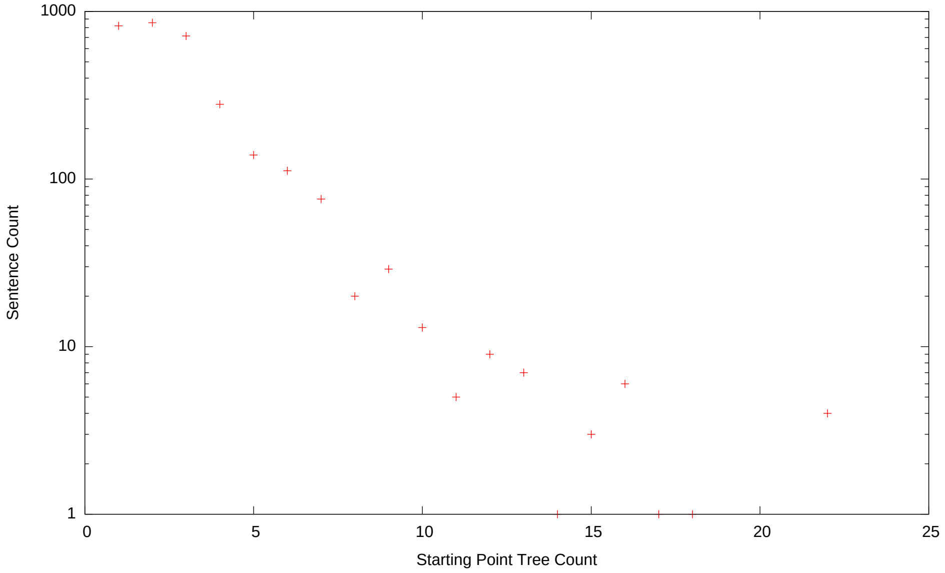


Figure 4.8: Distribution of Starting Point Tree Counts. Starting Point Tree Count is the number of starting point trees for a sentences. Sentence Count is the number of sentences from the DUC2003 test data that had that many starting trees.

of starting tree counts for the first five sentences of each document in the DUC2003 test data. This distribution shows that about 25% of sentences were unaffected by the multi-candidate rules, 65% generated 2 to 5 distinct starting point trees, and 10% generate 6 or more distinct starting-point trees, with a maximum of 22 distinct starting point trees.

Figure 4.9 shows the number of candidates generated for the entire DUC2003 test data using different sets of multi-candidate rules. The use of multi-candidate rules increases the number of distinct compressions. Figure 4.10 shows the candidate counts and Rouge-1 scores of Trimmer systems using different sets of multi-candidate rules. This graph shows that Rouge-1 scores generally increase as the size of the candidate set increases.

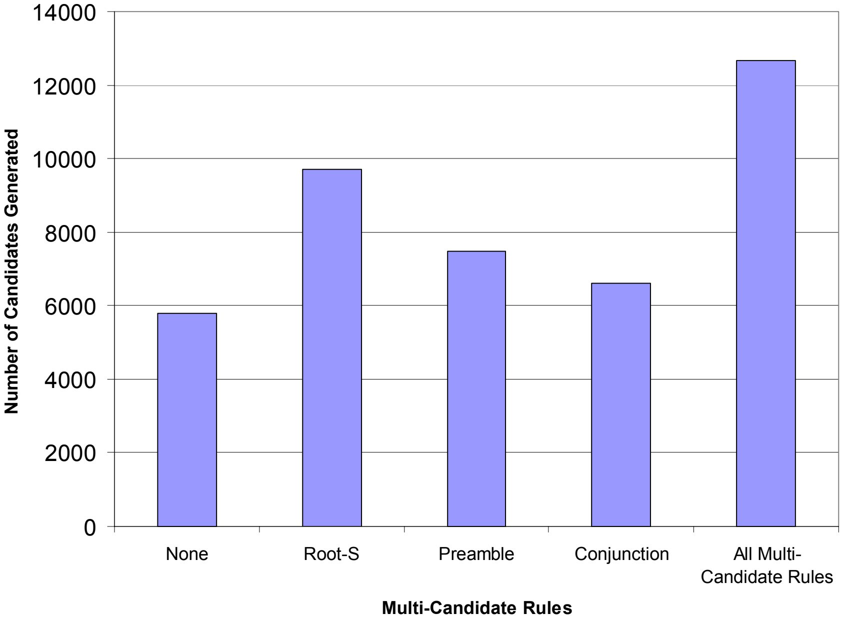


Figure 4.9: The number of candidates generated by Trimmer over the DUC2003 test set using different sets of multi-candidate rules. Greater use of multi-candidate rules increases the number of candidates.

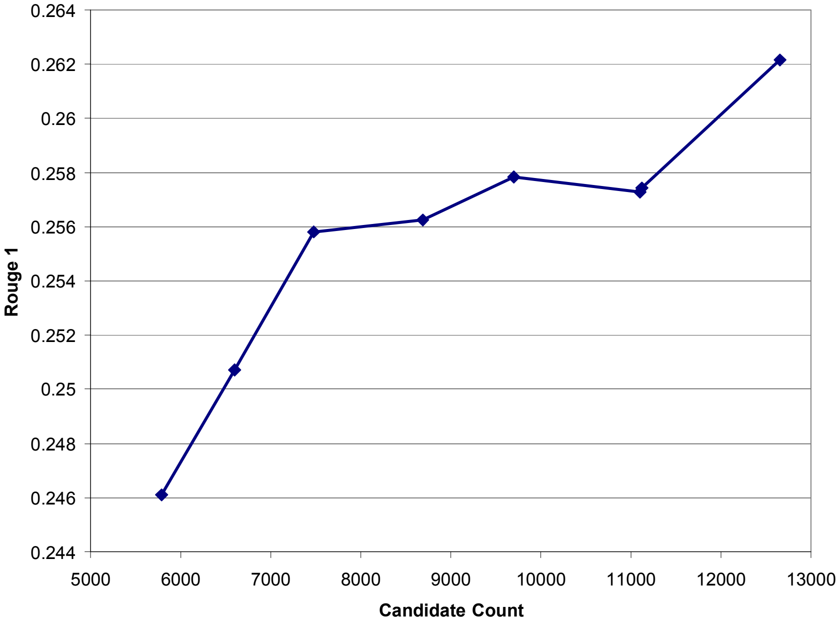


Figure 4.10: The candidate counts and ROUGE-1 scores for Trimmer on the DUC2003 test set using different sets of multi-candidate rules. Configurations of Trimmer that generate larger numbers of candidates tend to have higher ROUGE-1 scores.

4.4.2 Multiple Alternative Candidates with Topiary

Topiary makes use of the multiple-candidate approach by starting with the Trimmer compressions for a sentence. For each compression, Topiary creates additional versions by prepending each non-redundant topic term. If the resulting Topiary summary is below the length threshold, additional topics are added to fill the space. Note that the Trimmer compressions without any topics are preserved as possible Topiary summaries. The possible summaries provide a variety of balances between space devoted to topic terms and fluent text.

4.5 Sentence Selection for Single Document Summarization

The preceding section describes how Trimmer and Topiary are used to generate multiple compressed candidates for a source sentence. When multiple candidates are generated, a summarization system must also provide a means for selecting the best candidate to serve as the summary. This is the sentence selection component of the MASC framework.

The selection of the final summary from among the Topiary candidates is made using a linear combination of features. Trimmer uses three feature sets and Topiary uses an additional set of features related to topics.

The three Trimmer feature sets are:

- Length Features (L)
  - Number of characters
  - Number of words
- Rule Features (R)

  - Parse-tree depth of rule applications
  - Rule application counts
- Centrality (C)
  - Sentence to document centrality
  - Document to document set centrality

Length features (L) are observable features of the candidates. Rule features (R) include both depths and counts of linguistic rule applications during the Trimmer process. Centrality features (C) are measures of the similarity between two pieces of texts, derived from off-the-shelf information retrieval engines. Two such scores are computed: the similarity of a sentence to the document that contains it and the similarity of a document to its overall containing document set. For Topiary, the Trimmer features sets are augmented by two Topic features (T): the number of UTD topics and the sum of their scores. Trimmer and Topiary can perform selection using each feature set separately, and also in combinations for a total of 7 Topiary conditions (L, R, C, LR, LC, RC, and LRC) and 15 Topiary conditions (L, R, C, T, LR, LC, LT, RC, RT, CT, LRC, LRT, LCT, RCT, and LRCT). An additional condition—labeled None—refers to a run where no feature sets are used. In the absence of feature sets, the longest candidate that is under the length threshold is chosen as the summary.

Each feature set condition can be combined with each of the 8 multi-candidate rule conditions, making 64 possible Trimmer conditions and 128 possible Topiary conditions. Note that the condition in which no feature sets are used and no multi-candidate rules are used is equivalent to the single-candidate Trimmer and Topiary systems. In conditions where no feature sets are used, but some multi-candidate rules are enabled, Trimmer selects the longest candidate under the length threshold. There is no equivalently simple strategy to select a candidate for Topiary under these conditions, so Topiary will not be evaluated for the conditions where no feature sets are used but some multi-candidate rules are enabled.

The weights for features are set separately for each combination of feature sets and multi-candidate rules. The weights are optimized for Rouge-1 Recall on a training set. The BBN Optimizer, an implementation of Powell's Method (Powell, 1965), was used to perform the feature weight optimization.

4.6 Automatic Evaluations of Single Document Summarization

The single-document, single-candidate parse-and-trim approaches described in Sections 4.2 and 4.3.2 were implemented and submitted to the DUC2003 and DUC2004 automatic summarization evaluations using a fixed-order set of rule applications and, for Topiary, a requirement for two topic terms—an approach that generated a single candidate for each document. This section presents three evaluations of single document summarization systems. The first evaluation includes the performance of single candidate Topiary in the DUC2004 official evaluation and a post-hoc comparison with Topiary within the MASC framework. The second evaluation explores the effects of optimization feature sets and multi-candidate rules on MASC sentence selection. The third evaluation is an overview of all the single document summarization approaches of this dissertation on a common test data set.

4.6.1 Evaluation of Single-Candidate Topiary and MASC Topiary

The DUC2004 single document summarization task was to construct generic 75-byte summaries for 500 documents drawn from AP Newswire and the New York Times. The average size of the documents was 3,784 bytes, so a 75-byte summary represents a compression ratio of 2.0%.

Topiary was the highest scoring automatic system according to the official evaluation at DUC2004. Table 4.3 shows the official results from the DUC2004 evaluation. Topiary was system 6. System 1 was a baseline consisting of the first 75 characters of the document. Topiary was the only system to score significantly higher at 95% confidence than the baseline, and the only system to score higher than one of the human summary writers. The letters A through H denote the human summary writers.

Table 4.4 shows the Rouge scores of the DUC2004 Topiary submitted system (a single-candidate system) and the multi-candidate Topiary system.[7] This evaluation uses the same version and parameters for Rouge as the original DUC2004 evaluation.[8] The multi-candidate system scores higher than the single-candidate system for all Rouge measures, and the difference is significant for two of the measures: Rouge-2 and Rouge-L.

[7] The multi-candidate system was configured to use all multi-candidate rules, and all four features sets, with feature weights trained on the DUC2003 test data to optimize Rouge-1 recall.

[8] The version and configuration used for the DUC2004 evaluation was ROUGEeval-JK-1.2.1.pl -a -c 95 -b 75 -m -n 4 -w 1.2

System	R1	R2	R3	R4	RL	RW1.2
53	0.20423	0.04332	0.00881	0.00091	0.16264	0.09720
78	0.20564	0.00735	0.00017	0.00007	0.15039	0.08763
9	0.21151	0.02454	0.00221	0.00035	0.16516	0.09642
135	0.21265	0.05978	0.02017	0.00676	0.18685	0.11294
130	0.21668	0.02429	0.00263	0.00020	0.16738	0.09742
136	0.21736	0.06181	0.02070	0.00693	0.19109	0.11544
77	0.21945	0.05913	0.01775	0.00558	0.19357	0.11372
131	0.21948	0.02545	0.00335	0.00034	0.17608	0.10164
137	0.22101	0.06377	0.02134	0.00730	0.19443	0.11748
1	0.22136	0.06370	0.02118	0.00707	0.19411	0.11738
H	0.25017	0.06761	0.02162	0.00610	0.21402	0.12397
6	**0.25033**	**0.06501**	**0.02130**	**0.00717**	**0.20074**	**0.11957**
G	0.27184	0.06563	0.01955	0.00622	0.22850	0.13049
F	0.28369	0.06992	0.02282	0.00900	0.23208	0.13288
C	0.30009	0.08829	0.03036	0.01088	0.24597	0.14151
D	0.30416	0.09030	0.03021	0.01043	0.25654	0.14695
A	0.30647	0.09685	0.03498	0.01354	0.25970	0.14826
B	0.30767	0.09283	0.03332	0.01420	0.26393	0.15215
E	0.31478	0.10144	0.03893	0.01371	0.26630	0.15369

Table 4.3: Official DUC2004 Scores for Single-Document Summarization Task. There were 39 submissions to the evaluation. Shown here are the 8 human summaries, the baseline (System 1), and the top 10 scoring systems for Rouge1 Recall. Topiary is System 6.

Rouge Measure	**Topiary**	**MCR-based Topiary**
Rouge-1	0.25027	0.26684
Rouge-2	0.06484	0.08285
Rouge-3	0.02130	0.02853
Rouge-4	0.00717	0.00986
Rouge-L	0.20063	0.22538
Rouge-W1.2	0.11951	0.13150

Table 4.4: Comparison of DUC2004 submitted Topiary system and new MASC Topiary system. Differences are significant for Rouge-2 and Rouge-L.

4.6.2 Evaluation of Optimization Feature Sets and Multi-Candidate Rules on Trimmer and Topiary

Additional evaluations were performed on the single-candidate and multi-candidate versions of Trimmer and Topiary, to explore two areas of system configuration. First, different sets of features can be used for summary selection. Second, different combinations of multi-candidate rules, as described in Section 4.4.1, can be used in the generation of sentence compressions.

As described in Section 4.4.1, four distinct sets of features were used for sentence selection: length (L), rule application counts (R), topic counts and score (T), and centrality (C). These sets of features can be combined. For example, LRCT denotes the combination of all features; LT denotes a combination of length and topic features, etc. Three multi-candidate rules are available (Root-S, Preamble and Conjunction) and can be used alone or in combination.

The DUC2003 data set was used as the training data for this evaluation. Each combination of multi-candidate rules was used to generate a set of candidates for

Rules	**System**	**None**	**L**	**R**	**C**	**LR**	**LC**	**RC**	**LRC**
No-MC-Rules	Trimmer	0.203	0.230	0.220	0.232	0.241	0.239	0.215	**0.243**
	Topiary	0.251	0.238	0.245	0.242	0.252	0.249	0.257	**0.263**
Root-S	Trimmer	0.209	0.230	0.248	0.229	0.251	0.242	**0.253**	0.250
	Topiary	—	0.239	0.248	0.236	0.250	0.258	0.260	**0.262**
Preamble	Trimmer	0.186	0.231	0.220	0.234	**0.237**	0.238	0.218	0.229
	Topiary	—	0.235	0.247	0.234	0.254	0.248	0.256	**0.267**
Conjunction	Trimmer	0.198	0.231	0.230	0.228	0.225	**0.240**	0.218	0.228
	Topiary	—	0.238	0.247	0.240	0.249	0.249	0.256	**0.258**
Root-S/ Preamble	Trimmer	0.207	0.227	0.220	0.230	**0.245**	0.239	0.232	0.244
	Topiary	—	0.235	0.247	0.228	0.255	0.260	0.258	**0.269**
Root-S/ Conj	Trimmer	0.210	0.230	0.230	0.225	0.235	**0.242**	0.228	0.241
	Topiary	—	0.239	0.248	0.234	0.254	0.257	0.253	**0.261**
Preamble/ Conj	Trimmer	0.187	0.230	0.219	0.232	0.226	0.237	0.225	**0.238**
	Topiary	—	0.234	0.247	0.233	0.250	0.246	0.254	**0.263**
All-MC-Rules	Trimmer	0.209	0.227	0.208	0.227	0.237	0.238	0.209	**0.243**
	Topiary	—	0.234	0.246	0.227	0.254	0.258	0.255	**0.266**

Table 4.5: Rouge-1 Recall Scores for Topiary and Trimmer using different combinations of multi-candidate rules and different sets of features (excluding T) for candidate selection.

the documents of the DUC2003 data.[9]

The DUC2004 test data set was used as the evaluation data. The candidate sets were generated using each combination of multi-candidate rules. The summary for each document was selected using different combinations of feature sets, with the appropriate feature weights for that set of features and multi-candidate rules optimized on the DUC2003 data.

[9] This includes the combination when all multi-candidate rules are disabled and the candidates are generated solely by treating each intermediate application of a single-candidate rule as a candidate.

Rules	**T**	**LT**	**RT**	**CT**	**LRT**	**LCT**	**RCT**	**LRCT**
No-MC-Rules	0.220	0.251	0.260	0.221	0.258	0.259	0.259	**0.261**
Root-S	0.209	0.250	0.256	0.207	0.259	0.262	0.263	**0.267**
Preamble	0.201	0.243	0.262	0.249	0.267	0.259	0.261	**0.268**
Conjunction	0.221	0.249	0.258	0.221	**0.264**	0.258	0.263	0.262
Root-S/Preamble	0.193	0.244	0.258	0.242	0.264	0.265	0.262	**0.268**
Root-S/Conj	0.208	0.247	0.254	0.243	0.258	0.262	0.260	**0.267**
Preamble/Conj	0.200	0.243	0.258	0.243	0.265	0.259	0.261	**0.267**
All-MC-Rules	0.192	0.239	0.254	0.241	0.262	0.263	0.256	**0.268**

Table 4.6: Rouge-1 Recall Scores for Topiary using different combinations of multi-candidate rules and different sets of features (including T) for candidate selection.

Tables 4.5 and 4.6 show the Rouge-1 Recall scores of the Trimmer and Topiary system outputs on the DUC2004 test data using different sets of multi-candidate rules and optimized feature weights.[10] Table 4.5 shows feature combinations without topic-related features (no "T") and Table 4.6 shows feature combinations with topic-related features ("T"). In the conditions where neither topic terms nor feature sets were used (No-MC-Rules/None in Table 4.5), the Trimmer system selects a candidate by choosing the longest candidate under the length limit. This is essentially the same as applying trimming rules until a length threshold is reached. Topic-related features do not apply to the Trimmer system; thus, Trimmer does not appear in Table 4.6. The highest score in each row, i.e., across feature sets, is marked in bold.

The DUC2004 submitted system appears in Table 4.5 as a Topiary entry where no multi-candidate rules and no feature sets are used (No-MC-Rules/None)—with a

[10] The command used for the evaluation was ROUGE-1.5.5.pl -n 2 -m -u -c 95 -r 1000 -f A -p 0.5 -a -b 75.

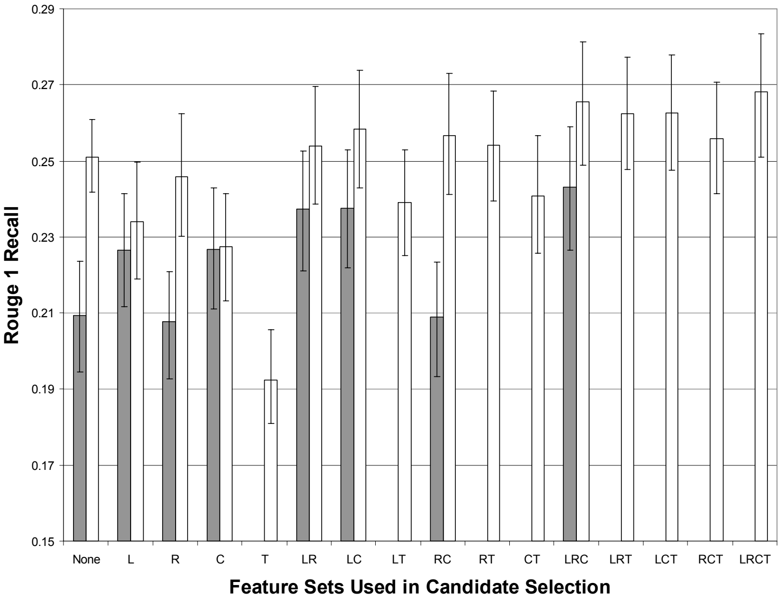


Figure 4.11: ROUGE-1 Recall evaluation of Topiary (white bars) and Trimmer (gray bars), varying by feature sets for candidate selection.

score of 0.25106. This system combined single-candidate sentence compression with topic terms. Because there is no simple Topiary analog for the use of Trimmer with multi-candidate rules in the absence of feature sets, seven conditions (under None) are left blank for Trimmer.

Figure 4.11 shows the scores for both systems using all three multi-candidate rules, with different combinations of features (None, L, R, C, T, LR, LC, etc.). The white bars correspond to the Topiary systems and the gray bars correspond to the Trimmer systems. The error bars indicate 95% confidence intervals. The Topiary systems outperform the Trimmer systems across the board. In addition, the MASC Topiary system using all four feature sets scores higher than the DUC2004 submitted (single-candidate) Topiary system, although the difference is not significant. In general, the systems that used larger combinations of feature sets got higher scores.

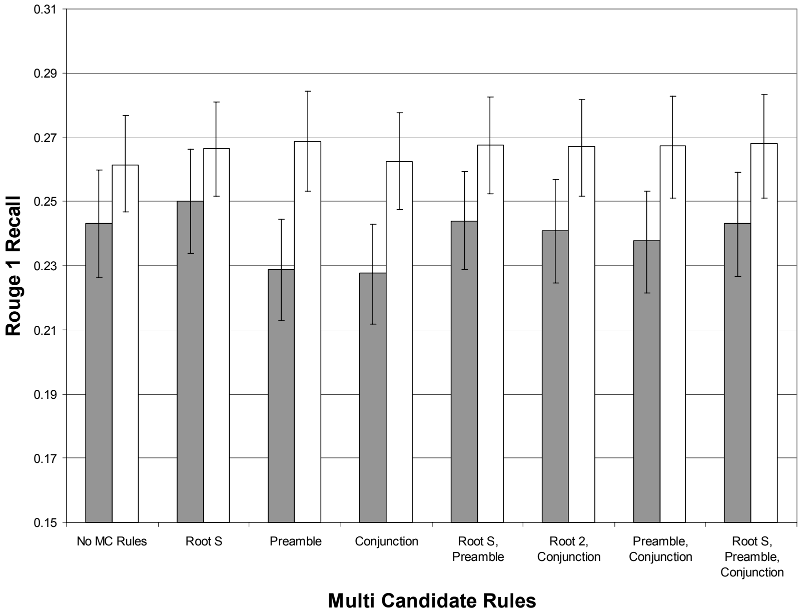


Figure 4.12: ROUGE-1 Recall evaluation of Topiary (white bars) and Trimmer (gray bars), varying by multi-candidate rules used in candidate generation.

In Tables 4.5 and 4.6 the high scores across feature sets tend toward the right, where larger numbers of feature sets are used.

Figure 4.12 shows another view of data, with different combinations of multi-candidate rules (No-MC-Rules, Root-S, Preamble, Conjunction, etc.), again using white bars for the Topiary systems and gray bars for the Trimmer systems. The Topiary systems use all four feature sets for candidate selection; the Trimmer systems use all features except "T." The figure shows that all Topiary systems that used multi-candidate rules to enlarge the pool of candidates scored higher than the systems that did not use any multi-candidate rules, although the difference is not significant.

Table 4.7 shows that when multi-candidate rules are used the number of candidates generated increases. Note, however, that increasing the size of the candidate

Multi-Candidate Rules	**Candidate Count**
No-MC-Rules (Intermediate Stage)	23522
Root-S	39250
Preamble	30963
Conjunction	28039
Root-S/Preamble	45812
Root-S/Conjunction	49998
Preamble/Conjunction	40086
All-MC-Rules	57909

Table 4.7: Count of Topiary Candidates over entire DUC2004 test set with different combinations of multi-candidate rules used in candidate generation.

pool does not by itself bring about an improvement in Rouge score. For example, Figure 4.12 shows that the Topiary system that uses only the Preamble multi-candidate rule achieves a higher score than the Topiary system that uses all three rules (even though the latter generates a larger number of candidates).

4.6.3 Overview Evaluation of Single-Document Summarization Systems

Table 4.8 presents a review of the systems proposed so far for performing the task of headline generation, or very short single-document summarization. HMM Hedge, Trimmer and Topiary are shown generating a single candidate, and within the MASC framework generating multiple candidates with feature-based selection among candidates. Trimmer and Topiary are shown with and without the use of multi-candidate rules (MCR). The UTD summaries consist of 75 characters of UTD

System	Rouge-1 Recall
First 75 Characters	**0.22564** (0.21059-0.24151)
UTD Topics	**0.21139** (0.19924-0.22349)
HMM Hedge Single Compression	**0.23090** (0.21620-0.24523)
HMM Hedge MASC	**0.26313** (0.24795-0.27780)
Trimmer Single Candidate	**0.20335** (0.18905-0.21843)
Trimmer Fixed-order, MASC	**0.24301** (0.22642-0.25985)
Trimmer MCR, MASC	**0.24305** (0.22667-0.25899)
Topiary Single Candidate	**0.25175** (0.23896-0.26446)
Topiary Fixed-order, MASC	**0.26144** (0.24678-0.27686)
Topiary MCR, MASC	**0.26810** (0.25101-0.28333)

Table 4.8: Review of single-document summarization systems, evaluated using Rouge-1.

topics, as described in Section 4.3.1. In addition a baseline consisting of the first 75 characters of each document is shown. The systems were run on the DUC2004 test data to produce 75-character single-document summaries. They were evaluated using Rouge-1.

This evaluation shows that sentence compression can be a useful component of an automatic summarization and that the MASC framework can significantly improve the value of compression in generating summaries. Single-candidate Topiary performs significantly better than single-candidate Trimmer and UTD. This supports the claim that single-document summaries combining topic terms and compressed text are better than corresponding systems consisting of topic terms alone or sentence compressions alone.

The highest scoring system is the Topiary MASC using multi-candidate rules.

HMM Hedge MASC and Topiary MASC score significantly higher than the baseline. Although HMM Hedge scores higher than Trimmer in Rouge evaluations, there are qualitative differences in the system outputs that affect the human usability of the systems, as described in the next section.

4.7 Human Evaluations of Trimmer, Topiary, and HMM Hedge

This section describes two human evaluations of single-document summarization systems using sentence compression. The first is a formal evaluation of Trimmer, Topiary, and HMM Hedge using an extrinsic task. The second is a human intrinsic evaluation and error analysis of Trimmer and Topiary.

4.7.1 Extrinsic Evaluation of Trimmer, Topiary, and HMM Hedge: Relevance Prediction

The formal extrinsic evaluation measured the ability of humans to decide whether a document is relevant to a query based on a 75-character summary. Both relevance judgments and relevance prediction were measured. The experiment shows that summarization can reduce the subjects' judgment time with some loss of accuracy, precision and recall. It also demonstrates a substantially different appraisal of HMM Hedge performance between the automatic evaluation and the human evaluation.

Six subjects were shown summaries for 20 documents for each of 20 topic descriptions. For each topic, half of the documents had been previously judged relevant to the topic by the Linguistic Data Consortium (LDC)[11], and half were judged non-relevant. Summaries from six sources were shown to the subjects: Single-candidate Trimmer, single-candidate Topiary, HMM Hedge, Human-written summaries, Newspaper Headlines, and First 75 Characters (baseline). Subjects were then shown the source documents and asked to make a relevance judgment after having read the full text. The interface used by the subjects timed how long it took them to make each judgment.

Sentences (31) through (33) show representative summaries from each system for three documents from this evaluation. Note that the HMM Hedge summaries contain many relevant terms, which accounts for HMM Hedge's high ROUGE scores, yet they are difficult to read and comprehend.

(31) Document APW19981016.0240

- (i) **HMM Hedge**: Cambodian leader Hun Sen opposition demands for accused of political crisis
- (ii) **Trimmer**: Cambodian leader Hun Sen rejected opposition parties' demands for talks out
- (iii) **Topiary**: PARTY Cambodian leader Hun Sen rejected opposition parties' demands for tal
- (iv) **Human**: Cambodian leader refuses to meet with opposition leaders outside Cambodia
- (v) **Newspaper Headline**: Hun Sen rejects opposition demands to hold talks outside Cambodia
- (vi) **First 75 Characters**: Cambodian leader Hun Sen on Friday rejected opposition parties' demands for

(32) Document APW19981129.0665

- (i) **HMM Hedge**: Albania says it uncovered by Islamic accused of bombings in parts of Europe
- (ii) **Trimmer**: Albania says it uncovered terrorist network operated by Osama Bin Laden, Is

[11] http://www.ldc.upenn.edu/

(iii) **Topiary**: BIN LADEN Albania says it uncovered terrorist network Laden, Islamic fundam

(iv) **Human**: Albania says terrorist network, linked to embassy bombing & BinLaden, uncovered

(v) **Newspaper Headline**: Report: Bin Laden operated terrorist network based in Albania

(vi) **First 75 Characters**: Albania says it has uncovered a terrorist network operated by Osama Bin Lad

(33) Document APW19981202.0581

(i) **HMM Hedge**: Dozens of Palestinians ambush car smashes with Israeli soldier and beat him

(ii) **Trimmer**: Dozens of Palestinians ambushed Israeli car, smashing windshields with sto

(iii) **Topiary**: ATTACK Dozens of Palestinians ambushed Israeli car, smashing windshields wi

(iv) **Human**: violence,shooting,stabbing in West Bank as Clinton plans to visit Israel

(v) **Newspaper Headline**: Israeli, Palestinians clash near West Bank settlement

(vi) **First 75 Characters**: Dozens of Palestinians ambushed an Israeli car on Wednesday, smashing its w

Two evaluations were derived from the experiment for each system: agreement with LDC and relevance prediction. Agreement with LDC measures how often the subjects made the same relevance judgment as the LDC annotators. Inter-annotator agreement on relevance judgments, even on full documents, is often very low. This can mean that the variation between subjects can swamp the effects of system differences. Relevance prediction addresses this problem by measuring the subjects' ability to anticipate their own judgments on full documents using summaries. The motivation behind relevance prediction is that since subjects will generally agree with themselves about the relevance of a document to a topic, the variation observed will be more likely to come from differences in the systems.

For both evaluations, three scores were calculated: Accuracy (A), Precision (P) and Recall (R). These scores are based on counts of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). Accuracy is ratio of true judgments to total judgments. Precision is the percentage of relevant judgments that match the LDC judgments. Recall is the percentage of LDC relevant documents that the subject judged relevant. For relevance prediction, the subject's own judgments on the full documents take the place of the LDC relevance judgments.

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

LDC agreement is shown in Table 4.9, along with the average time the subjects spent on each document. Relevance prediction is shown in Table 4.10. Note that FullText does not appear in this table, because for relevance prediction, FullText is the gold standard.

This experiment shows that relevance judgments can be made much more quickly using summarizations, with some loss of accuracy, precision and recall. Note that even the summaries written by humans, Headline and Human, show a small drop in the scores. Automatic systems have higher precision scores than recall scores. They are more likely to cause a subject to miss a relevant document than to select a non-relevant one. With the exception of HMM Hedge Recall, all of the automatic systems are performing higher than chance (.50), but none outperforms

System	A	P	R	Time(sec)
Headline	.83	.83	.83	7.0
Human	.78	.80	.75	6.9
Topiary	.76	.84	.63	5.9
First75	.76	.84	.63	7.6
Trimmer	.76	.83	.65	5.7
HMM	.68	.72	.57	6.9
FullText	.87	.86	.89	159.9

Table 4.9: LDC Agreement Scores, showing Accuracy, Precision, Recall and Average time per document, in seconds.

System	A	P	R
Headline	.82	.87	.79
Human	.88	.88	.86
Topiary	.76	.91	.62
First75	.79	.96	.65
Trimmer	.79	.83	.70
HMM	.63	.72	.52

Table 4.10: Relevance Prediction Scores, showing Accuracy, Precision and Recall

the baseline of First75. The low scores for HMM Hedge in both LDC agreement and relevance prediction support the interpretation that although HMM Hedge scores well on Rouge in comparison to Trimmer, the poor readability of HMM Hedge summaries make them less usable for extrinsic tasks than Trimmer summaries. The discrepancy between the automatic evaluation results and extrinsic evaluation of HMM indicates that automatic evaluation of summarization is still a controversial area. Research in the area of correlating automatic evaluation with human evaluation includes Papineni et al. (2002) and Lin and Hovy (2003) and Hobson et al. (2007).

Follow-on work for this study could involve inspecting the summaries to determine if there are any problems typical of the FN and FP summaries, such as low number of query terms, unresolved anaphora or ungrammaticality.

4.7.2 Human Intrinsic Evaluation: Trimmer and Topiary

A human intrinsic comparison of Topiary and Trimmer using multi-candidate generation with selection based full feature sets was performed on the system output of 100 75-character single-document summaries from the DUC2004 test data. The comparison revealed that in 54 cases the Topiary summary was more informative and correct than the Trimmer summary, in 19 cases the Trimmer summary was more informative and correct, and in 27 cases the summaries were of equivalent quality.

When Topiary succeeds, it does so by including a topic term that provides vital context for an otherwise ambiguous sentence, as in Sentence (34i). When it does not succeed it is generally because the topic term doesn't help establish context, as in Sentence (35i), or because the topic term is misleading, as in Sentence (36i).

The set of documents containing Sentence (36i) is the result of a search for documents dealing with an Asian-Pacific Economic Cooperation meeting held in Malaysia. Seven of ten stories mention the effect on the conference of international reaction to the prosecution of Malaysian deputy prime minister Anwar Ibrahim on morals charges. Topiary misleadingly gives ANWAR as a topic term for two of the remaining stories. This type of error occurs when a concept is central to the majority of stories in a document set and appears in the search terms, but is entirely absent from some stories in the set.

(34) (i) Topiary: PINOCHET he is too sick to be extradited to Spain to face charges of genoci

(ii) Trimmer: wife of former Chilean dictator Augusto Pinochet appealed, saying he is to

(35) (i) Topiary: PEOPLE Honduras braced for potential catastrophe as Hurricane Mitch roared

(ii) Trimmer: Honduras braced as Hurricane Mitch roared, churning up high waves and inte

(36) (i) Topiary: ANWAR Taiwan will send chief economic planner to represent President Lee Te

(ii) Trimmer: Again bowing, Taiwan will send chief economic planner to represent Preside

4.8 Summary

This chapter has described how the underlying select-words-in-order approach to sentence compression for single-document summarization was implemented by using syntactic rules based on linguistic comparisons of Headlinese and general English. Trimmer rules and algorithms were presented along with Topiary, a hybrid topic term and fluent text single document summarization system. Trimmer and Topiary were enhanced to produce multiple alternative sentence compressions and use linear combinations of feature scores to select the best candidate as a document summary.

An evaluation with ROUGE demonstrated that Topiary outperforms Trimmer and HMM Hedge, and that the generation of multiple candidates with feature-based sentence selection improves Topiary beyond its state-of-the-art performance in DUC2004. The use of multi-candidate rules was found to have promise, although it did significantly improve ROUGE scores. ROUGE-1 recall generally increased as the number of feature sets used in optimization increased. An automatic evaluation on a common data set of all the single-document summarization systems presented thus far showed that sentence compression is a useful tool in performing automatic summarization. Finally a human evaluation using relevance prediction was performed on Trimmer, Topiary, and HMM Hedge that showed subjects were able to use automatic summarization output to perform a relevance judgment task much faster than they could with the full texts, but at a cost in precision and recall. Notably, automatic evaluation tools give much higher relative ranking to HMM Hedge summarizations than the human evaluation. An intrinsic comparison of Trimmer and Topiary revealed that Topiary usually generates better summaries than Trimmer, and when Topiary doesn't succeed it is due to uninformative or misleading topic selection.

Chapter 5

Multi-Document Summarization

The sentence compression tools presented for single-document summarization have been incorporated into the MASC framework for multi-document summarization, shown in Figure 5.1. Multi-document summarization is the generation of a summary for a collection of documents rather than a single document. The summaries should cover the information from the collection which is most important, or most relevant to an information need. Multi-document summaries typically contain multiple sentences, so both anti-redundancy and information coverage are important goals.

MuD (Multi-Document) MASC produces a textual summary from a collection of relevant documents in three steps. First, sentences are selected from the source documents for compression in the filtering step. The most important information occurs near the front of the stories, so the first five sentences are selected from each document for compression in this step. Second, multiple compressed versions of each sentence are produced using Trimmer or HMM Hedge to create a pool of candidates for inclusion in the summary. Finally, a sentence selector constructs the summary by iteratively choosing from the pool of candidates based on a linear combination of features until the summary reaches a desired length.

Multi-document summarization differs from very short single-document sum-

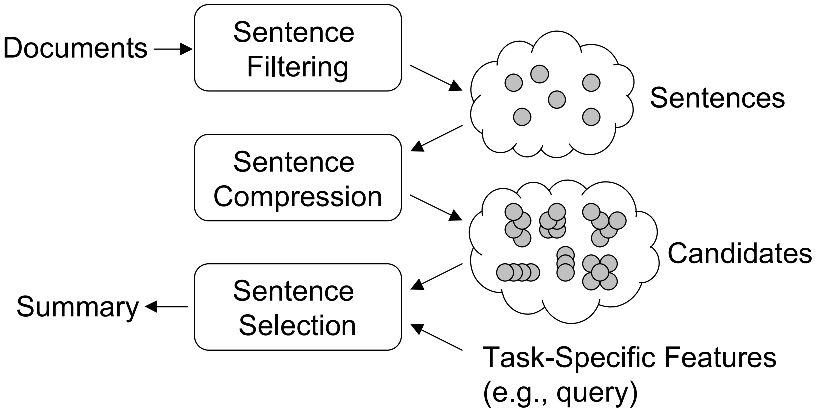


Figure 5.1: The Multiple Alternative Sentence Compression (MASC) framework for automatic summarization

marization with respect to sentence compression in that the length constraint is global over several sentences rather than specific to a single sentence. Reducing the size of selected sentences while preserving their relevant content can allow space for the inclusion of additional sentences. However, given the interaction of relevance and redundancy in a multi-sentence summary, no single trimming algorithm or scoring metric can provide the best compression of a sentence in advance. For example consider the two source sentences in example (37).

(37) (i) Source: Gunmen killed a prominent Shiite politician.

(ii) Source: The killing of a prominent Shiite politician by gunmen appeared to have been a sectarian assassination.

(iii) Compressed: The killing appeared to have been a sectarian assassination.

Suppose that Sentence (37i) has already been selected for inclusion in the summary. Sentence (37ii) contains some redundant information and some novel information. Sentence (37iii) is a compression of Sentence (37ii) in which the prepositional phrases "of a prominent Shiite politician" and "by gunmen" have been trimmed. Even though these prepositional phrases contain important information from the perspective of a generic summary, they are redundant to a summary containing Sentence (37i). By generating many alternative sentence compressions in the compression phase, the MuD MASC framework provides the sentence selection phase with a wider variety of extracted sentences. Without multiple compressions, the sentence selection phase would be limited to the source sentences, their single most likely compressions (HMM Hedge), or the results of a deterministic algorithm (Trimmer).

The first section of this chapter describes a human annotation task that estimates the potential space-saving value of sentence compression for a multi-document summarization system. The chapter continues with a description of the components of MuD MASC and addresses HMM-based and Trimmer-based MuD MASC. It will also describe how Trimmer was used as a component in a collaborative multi-document summarization system developed with IDA/CCS (Conroy et al., 2005, 2006b). An implementation-level description of the MASC sentence selector is given in Appendix D.

5.1 Potential Benefit of Sentence Compression for Multi-Document Summarization

MuD MASC aims to improve summaries by using compression to make room within the summary length constraint for additional information. The approach assumes that relevant sentences from the source documents can be compressed in such a way that they remain relevant and take up less space. A preliminary study was performed to confirm this assumption. One subject was shown 103 sentences, each of which had already been judged relevant to one of 39 queries. The subject was asked to make relevance judgments for 430 compressed versions of the original relevant sentences, produced by Trimmer. In other words, the subject decided whether a compressed version of a query-relevant sentence was also relevant. 68% of the sentences had at least one compressed version that was also relevant to the query.

To measure the greatest possible compression while preserving relevance, two collections of sentences were assembled. The first consisted of the original relevant sentences, the second consisted of the shortest compression of each sentence that was judged relevant or the original sentence if no compression had been judged relevant.

The sizes in words and characters of the two collections are shown in Table 5.1. Suppose that a summary is made for each of the 39 queries that consists of the sentences that are relevant to the query. The combined length of the summaries will be 3190 words. If the summaries consist instead of the shortest relevant compression of the relevant summaries, the combined length will be 2576 words. This study shows the potential for sentence compression to give a 16.7% reduction by word count or 17.6% reduction by character count over the collection of summaries. For the task of making a 250 word summary, this space saving would permit the inclusion of 4 additional 10-word sentences in the summary.

Sentence Collection	Size in Words	Size in Characters
Original	3190	19768
Shortest Relevant	2657	16286

Table 5.1: Size in words and characters of original sentences and shortest relevant compressions.

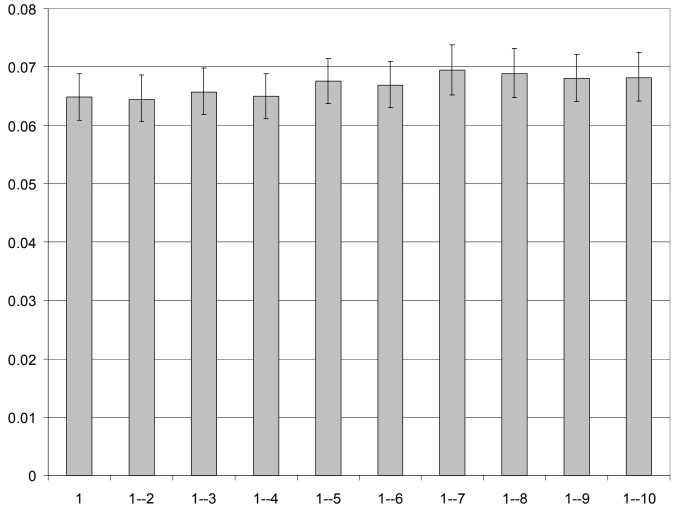


Figure 5.2: Multi-document summarization, filtering increasing numbers of sentences from the start of the document.

5.2 MuD MASC Filtering

In very short single document written news summarization it is almost always the case that the first sentence is the best sentence to select for compression. In the case of multi-document summarization it is not obvious that this pattern will hold. If multiple documents are about the same topic, always choosing the lead could result in redundancy. Figure 5.2 shows a Rouge evaluation of a multi-document summarization system with no bias for sentence position in which the MASC filtering increases the windows of sentences from only the first sentences to the first

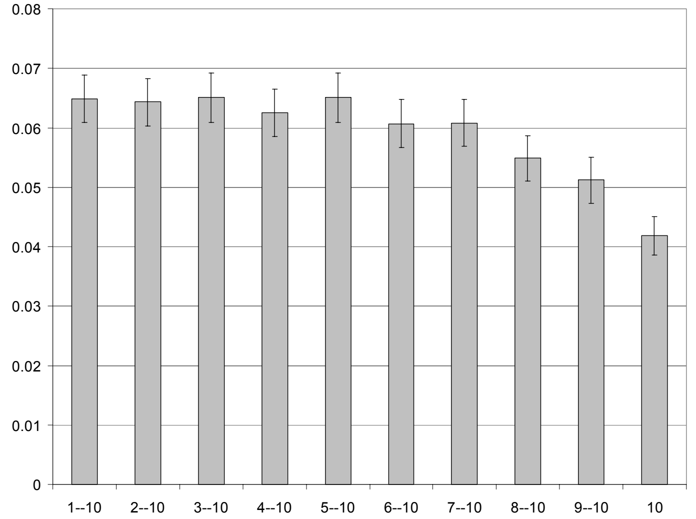


Figure 5.3: Multi-document summarization, filtering increasing numbers of sentences from the start of the document.

through tenth sentences. The scores indicate a non-significant increase as additional sentences are allowed through the filtering stage.

However this does not necessarily mean that sentence position is irrelevant to multi-document summarization. Figure 5.3 shows a similar progression in which the range of sentences allowed through the filter decreases by removing sentences from the front. The scores decrease significantly as sentences from the starts of documents are filtered out. This initial investigation of the effect of filtering on multi-document summarization indicates that a bias for sentences early in documents remains, but it is not as strong as in the single-document short summary task. Throughout this section filtering will consist of selecting the first five sentences of each document unless otherwise specified.

5.3 MuD MASC Sentence Selection

The sentence selector produces a multi-document summary by scoring each sentence using a linear combination of features. The implementation of the sentence selector is based on the idea of Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998), in which relevance and anti-redundancy are balanced. All the candidates are given a score which is a linear combination of relevance, anti-redundancy and compression-specific features. The weights of the features are chosen to maximize the ROUGE-2 recall score over a corpus of training data, because this metric has been shown to have a good correlation with human judgment and is the accepted evaluation metric for this task in the summarization community (Lin, 2004; Dang and Harman, 2006).

The highest ranking sentence from the pool of eligible candidates is chosen for inclusion in the summary. When a candidate is chosen, all other compressed variants of that sentence are eliminated. After a sentence is chosen, the dynamic features are re-calculated, and the candidates are re-ranked. This process is repeated until the length of the summary in words is greater than or equal to the length threshold. The ordering of the sentences within the summary is described in Section 5.3.3.

5.3.1 Static Features

Static features are calculated before sentence selection begins, and do not change during the process of summary construction:

- Position. The zero-based position of the sentence in the document.
- Sentence Relevance. The relevance score of the sentence to the query.

- Document Relevance. The relevance score of the document to the query.
- Sentence Centrality. The centrality score of the sentence to the topic cluster.
- Document Centrality. The centrality score of the document to the topic cluster.
- Scores from Compression Modules:
  - Trim rule application counts. For Trimmer-based MCR, the number of trimmer rule instances applied to produce the candidate.
  - Negative Log Desirability. For HMM-based MCR, the relative desirability score of the candidate.

Some features are derived from the sentence compression modules used to generate candidates. For Trimmer, the rule application count feature of a candidate is the number of rules that were applied to a source sentence to produce the candidate. The rules are not presumed to be equally effective, so the rule application counts are broken down by rule type. HMM Hedge uses the relative desirability score calculated by the decoder, expressed as a negative log.

The Uniform Retrieval Architecture (URA), University of Maryland's software infrastructure for information retrieval tasks, is used to compute relevance and centrality scores for each compressed candidate. There are four such scores: the relevance score between a compressed sentence and the query, the relevance score between the document containing the compressed sentence and the query, the centrality score between a compressed sentence and the topic cluster, and the centrality score between the document containing the compressed sentence and the topic cluster. The topic cluster is defined to be the entire collection of documents relevant to this particular summarization task. Centrality is a concept that quantifies how similar a piece of text is to all other texts that discuss the same general topic. Sentences having higher term overlap with the query and sources that are more “central” to the topic cluster are preferred for inclusion in the final summary.

The relevance score between a compressed sentence and the query is an *idf*-weighted count of overlapping terms (number of terms shared by the two text segments). Inverse document frequency (*idf*), a commonly-used measure in the information retrieval literature, roughly captures term salience. The *idf* of a term i is defined by $log(N/df_i)$, where N is the total number of documents in a particular corpus, and df_i is the number of documents containing term i; these statistics were calculated from one year’s worth of LA Times articles. Weighting term overlap by inverse document frequency captures the intuition that matching certain terms is more important than matching others.

Lucene, a freely-available off-the-shelf information retrieval system, is used to compute the three other scores. The relevance score between a document and a query is computed using Lucene’s built-in similarity function. The centrality score between the candidate and a topic cluster is the mean of the similarity between the candidate and each document comprising the cluster (once again, as computed by Lucene’s built-in similarity function). The document-cluster centrality score is also computed in much the same way, by taking the mean of the similarity of the particular document with every other document in the cluster. In order to obtain an accurate distribution of term frequencies to facilitate the similarity calculation, an index was made of all relevant documents (i.e., the topic cluster) along with a comparable corpus (one year of the LA Times)—this additional text essentially serves as a background model for non-relevant documents.

5.3.2 Dynamic Features

Dynamic features change during the process of sentence selection to reflect changes in the state of the summary as sentences are added.[1] The dynamic features are:

- Redundancy. A measure of how similar the sentence is to the current summary.
- Sentence-from-doc. The number of sentences already selected from the sentence's document.

The intuition behind the redundancy measure is that candidates containing words that occur much more frequently in the current state of the summary than they do in general English are redundant to the summary. The measure supposes that sentences in the summary are generated by the underlying word distribution of the summary rather than the distribution of words in the general language. If a sentence appears to have been generated by the summary rather than by the general language, it is considered to be redundant to the summary. Consider a summary about earthquakes. The presence in a candidate of words like *earthquake*, *seismic*, and *Richter Scale*, which have a high likelihood in the summary, will make us think that the candidate is redundant to the summary.

The redundancy measure estimates the extent to which a candidate is more likely to have been generated by a summary than by the general language using the probabilities of the words in the candidate. The estimate of the probability that a

[1] At present the dynamic features are properties of the candidates, calculated with respect to the current summary state. There are no features directly relating to the amount of space left in the summary, so there is no mechanism that would affect the distribution of compressed candidates over the iterations of the sentence selector. This issue will be addressed as future work in Section 7.2.

word w occurs in a candidate generated by the summary is

$$P(w) = \lambda P(w|D) + (1 - \lambda)P(w|C)$$

where D is the summary, C is the general language corpus[2], λ is a parameter estimating the probability that the word was generated by the summary, and $(1 - \lambda)$ is the probability that the word was generated by the general language. As a general estimate of the portion of words in a text that are specific to the text's topic, λ has been set to 0.3.[3] The probabilities are estimated by counting the words[4] in the current summary and the general language corpus:

$$P(w|D) = \frac{count\ of\ w\ in\ D}{size\ of\ D}$$

$$P(w|C) = \frac{count\ of\ w\ in\ C}{size\ of\ C}$$

The probability of a sentence is taken to be the product of the probabilities of its words, so the probability that a sentence was generated by the summary, i.e. the redundancy metric, is calculated by:

$$Redundancy(S) = \prod_{s \in S} \lambda P(s|D) + (1 - \lambda)P(s|C)$$

Log probabilities are used for ease of computation:

$$\sum_{s \in S} \log(\lambda P(s|D) + (1 - \lambda)P(s|C))$$

Redundancy is a dynamic feature because the word distribution of the current summary changes with every iteration of the sentence selector.

[2] The documents in the set being summarized are used to estimate the general language model.

[3] In the future, the value of λ will be optimized to indirectly maximize the performance of the summarizer by using $P(s|D)$ and $P(s|C)$ as separate dynamic features, and observing the relationship between the learned weights for these two features.

[4] Actually, preprocessing for redundancy includes stopword removal and applying the Porter Stemmer (Porter, 1980).

5.3.3 Sentence Ordering

Once the sentences have been selected for inclusion in the summary the question remains how they should be ordered in the summary. Two approaches have been tried. The first is to group the sentences by source document and sort by document-position within documents. The second is to present the sentences in the order that they were selected for inclusion in the summary. The second approach achieves non-significantly higher Rouge scores because of final sentence truncation. Sentence truncation is required because of the hard limit on the number of words imposed by the multi-document summarization task description, presented in Section 5.4.3. When sentences are grouped by document it is possible to truncate highly relevant and non-redundant sentences. However when the sentences are sorted by selection order, the truncated sentence will always be the least relevant and/or most redundant sentence in the summary. The systems presented in this chapter use selection order. In future work, the compression tools and MASC framework will be combined with more sophisticated sentence ordering, such as that used by Conroy et al. (2006b).

5.4 Multi-Document Summarization Systems

MuD MASC can be used in combination with HMM Hedge or Trimmer sentence compressions. Trimmer-based MuD MASC was submitted to the DUC2006 evaluation. After the evaluation, an HMM-based MuD MASC was evaluated on the DUC2006 test data. This section will describe how HMM Hedge and Trimmer are used as input to MuD MASC. The results of the post-hoc evaluation on DUC2006 test data are discussed in Section 5.4.4.

5.4.1 Multi-Document Summarization with HMM Hedge

HMM Hedge was used to generate 55 candidate compressions for the 10 most relevant sentences (as determined by URA) from each document, and the source sentence was also included as a candidate. A compression-specific feature was used with value 0 if the candidate is an original source sentence, and value 1 if the candidate was compressed by HMM Hedge.

5.4.2 Multi-Document Summarization with Trimmer

Trimmer applies many trimming rules to a sentence in the process of compressing a sentence to the desired size. There are a large number of possible trimmed compressions of a sentence. In order for Trimmer to interact well with MuD MASC Trimmer must ensure that the compressions it provides to the selector support the selection of relevant, non-redundant variants. Consider a sentence with two important pieces of information, one of which is redundant to the summary. Trimmer cannot know in advance which piece of information will be non-redundant, so it should provide both pieces separately as alternate compressions.

Trimmer was used to create multiple candidates for the sentences of the source documents. The original source sentences were also included as a candidates. The number of Trimmer rules applied to the sentence was used as a compression-specific

Title: Native American Reservation System—pros and cons
Narrative Description: Discuss conditions on American Indian reservations or among Native American communities. Include the benefits and drawbacks of the reservation system. Include legal privileges and problems.

Figure 5.4: Topic D0601A from the DUC2006 multi-document summarization task.

feature. Feature weights were set by manually optimizing the Rouge-2 average recall on DUC2005 test data.

5.4.3 Examples of System Output

The MASC framework was applied to test data from the DUC2006 evaluation (Dang and Harman, 2006). Three systems were tried, using HMM Hedge, Trimmer and no-compression as the compression stage. The Trimmer-based system was submitted to the official evaluation. Given a topic description and a set of 25 documents related to the topic (drawn from AP newswire, the New York Times, and the Xinhua News Agency English Service), a system's task was to create a 250-word summary that addressed the information need expressed in the topic. One of the topic descriptions is shown in Figure 5.4. The 25 documents in the document set have an average size of 1170 words, so a 250-word summary represents a compression ratio of 0.86%.

Figures 5.5, 5.6 and 5.7 show examples of MASC output using Trimmer compression, HMM Hedge compression, and no compression. For readability, ○ is used as a sentence delimiter; this is not part of the actual system output. The HMM Hedge compressions do not contain punctuation because punctuation tokens are

Seeking to get more accurate count of country's American Indian population, Census Bureau turning to tribal leaders and residents on reservations to help overcome long-standing feelings. ○ American Indian reservations would get infusion. ○ Smith and thousands seeking help for substance abuse at American Indian Community House, largest of handful of Native American cultural institutions. ○ Clinton going to Pine Ridge Reservation for visit with Oglala Sioux nation and to participate in conference on Native American homeownership and economic development. ○ Said Glen Revere, nutritionist with Indian Health Services on 2.8 million-acre Wind River Reservation, about 100 miles east of Jackson, Wyo. "Then we came up with idea for this community garden, and it been bigger than we ever expected." ○ Road leading into Shinnecock Indian reservation is not welcoming one But main purpose of visit – first to reservation by president since Franklin Roosevelt – was simply to pay attention to American Indians, who raked by grinding poverty Clinton's own advisers suggested he come up with special proposals geared specifically to Indians' plight. ○ "This highlights what going on out there, since beginning of reservation system," said Sidney Harring, professor at City University of New York School of Law and expert on Indian crime and criminal law. ○ American Indians are victims. ○ President Clinton turned attention to arguably poorest, most forgotten ○ U.S. citizens: American Indians. ○ When American Indians began embracing gambling, Hualapai tribe moved quickly to open casino. ○ members of Northern Arapaho Tribe on Wind River Reservation started seven-acre community garden with donated land, seeds and

Figure 5.5: MuD MASC Summary for DUC2006 Topic D0601A, using Trimmer for sentence compression.

David Rocchio deputy legal counsel to Vermont Gov. Howard Dean who has been involved in discussions on Indian gambling through the National Governors Association said that the concern that governors have is not with the benefit casinos bring to tribes ○ Native Americans living on reservations that maintain 50 percent or more unemployment are exempt from the national five year family limit on welfare benefits ○ Smith and thousands like her are seeking help for their substance abuse at the American Indian Community House the largest of a handful of Native American cultural institutions in the New York area ○ Juvenile crime is one strand in the web of social problems facing urban and reservation Indian communities the report said ○ Soldierwolf's family represents the problems that plague many of the 1.3 million American Indians who live on reservations of whom 49 percent are unemployed ○ Powless said the Onondaga people want to work with the community outside the reservation to improve the economy of the region perhaps creating tourism destinations that might include Indian culture or setting up a free trade zone at unused manufacturing sites ○ As Indian communities across the nation struggle with short funds and a long list of problems they are watching the Navajo Nation's legal battle with the federal government ○ recognize Indians not only Native Americans as Americans ○ go on reservation system Harring Indian ○ Supreme Court allows legalized gambling Indian reservations ○ American Indian reservations tribal colleges rise faster than ○ main purpose of reservation to pay American Indians by poverty proposals

Figure 5.6: Mud MASC Summary for DUC2006 Topic D0601A, using HMM Hedge for sentence compression

Seeking to get a more accurate count of the country's American Indian population, the Census Bureau is turning to tribal leaders and residents on reservations to help overcome long-standing feelings of wariness or anger toward the federal government. ○ American Indian reservations would get an infusion of $1.2 billion in federal money for education, health care and law enforcement under President Clinton's proposed 2001 budget ○ Smith and thousands like her are seeking help for their substance abuse at the American Indian Community House, the largest of a handful of Native American cultural institutions in the New York area. ○ Clinton was going to the Pine Ridge Reservation for a visit with the Oglala Sioux nation and to participate in a conference on Native American homeownership and economic development. ○ said Glen Revere, a nutritionist with the Indian Health Services on the 2.8 million-acre Wind River Reservation, about 100 miles east of Jackson, Wyo. "Then we came up with the idea for this community garden, and it's been bigger than we ever expected in so many ways." ○ The road leading into the Shinnecock Indian reservation is not a welcoming one ○ But the main purpose of the visit – the first to a reservation by a president since Franklin Roosevelt – was simply to pay attention to American Indians, who are so raked by grinding poverty that Clinton's own advisers suggested he come up with special proposals geared specifically to the Indians' plight. ○ "This highlights what has been going on out there for 130 years,

Figure 5.7: MuD MASC Summary for DUC2006 Topic D0601A, with no sentence compression

not included in the language models or the Viterbi decoder. In single document summaries, punctuation takes up space and does not detract much from readability. In multi-sentence summaries, however, punctuation contributes to readability, so future work on multi-document summarization using HMM Hedge compression should include post-processing tools to restore punctuation.

The sentences compressed by Trimmer mimic Headlinese by omitting determiners and auxiliary verbs. For example, the first sentence in Figure 5.5 is a compression of the following source sentence:

> Seeking to get a more accurate count of the country's American Indian population, the Census Bureau is turning to tribal leaders and residents on reservations to help overcome long-standing feelings of wariness or anger toward the federal government.

Three determiners and a form of *be* have been removed from the source sentence in the compression that appears in the summary. The removal of this material make the sentence appear more like a headline.

In comparison with Trimmer compressions, HMM compressions are generally less readable and more likely to be misleading. Consider the final sentence in Figure 5.6.

(38) main purpose of reservation to pay American Indians by poverty proposals

This is a compression of the following source sentence:

(39) But the main purpose of the visit—the first to a reservation by a president since Franklin Roosevelt—was simply to pay attention to American Indians, who are so raked by grinding poverty that Clinton's own advisers suggested he come up with special proposals geared specifically to the Indians' plight.

Because HMM Hedge uses a bigram model of Headlines, it is unable to capture sentence-level grammaticality. The same limitation makes it difficult to prevent misleading or incorrect compressions.

For example, the third sentence from the end of Figure 5.6 seems to say that a court legalized gambling on Indian reservations:

(40) Supreme Court allows legalized gambling Indian reservations

However, it is a compression of the following source sentence:

(41) Only Monday, the California Supreme Court overturned a ballot measure that would have allowed expansion of legalized gambling on Indian reservations.

Another advantage of sentence compression is that it can avoid truncation of a final sentence. DUC2006 topic D0602B deals with steroid use by female athletes. A MASC summarization system that did not use compression selected 8 sentences for the summary with the final sentence truncated as shown in Sentence (42i). A MASC system that used Trimmer for compression selected compressed versions of the same 8 sentences, but the compression created room in the summary for a fuller version of the truncated sentence, shown in Sentence (42ii) as well as an entire additional sentence, Sentence (43).

(42) (i) Truncated: All five defendants denied, however,

(ii) Compressed: All five defendants denied, however, that they knowingly damaged health of athletes, some of whom were as young as when they started taking pink and blue anabolic steroid pills.

(iii) Source: All five defendants denied, however, that they knowingly damaged the health of the athletes, some of whom were as young as 12 when they started taking the pink and blue anabolic steroid pills.

(43) Manfred Ewald, 74, and Manfred Hoeppner, were each convicted of 20 counts of being accessories to causing bodily harm

The examples show that sentence compression allows a summary to include more material from other sources. This increases the topic coverage of system output.

5.4.4 Evaluation of Multi-Document Summarization Systems

The official evaluation at DUC2006 included Rouge-1 and Rouge-2 average recall and precision. Rouge is described in Section 2.3.1. For DUC2006, four model summaries were provided for each document cluster. For consistency with the DUC2006 evaluation, jackknifing, described in Section 2.3.1 is used with Rouge in the evaluation that follows. However, this evaluation differs from the official DUC2006 evaluation in that Rouge is configured to omit stopwords from the calculation. The removal of non-essential stopwords (typical of Headlinese) is an important component of Trimmer-based sentence compression.

Trimmer and HMM Hedge were evaluated as compression components within the MuD MASC summarization framework, along with a baseline that uses the same sentence selector but does not use any sentence compression. All systems performed the filtering stage by selecting the first five sentences of each document. The final stage of sentence selection was performed using an MMR-based sentence selector, as described in Section 5.3. The feature weights were manually optimized to maximize Rouge-2 recall on a comparable training corpus, 1,593 Financial Times and Los Angeles Times articles grouped into 50 topics from the DUC2005 query-focused multi-document summarization test data. The systems were used to generate query-focused, 250-word summaries using the DUC2006 test data, described in Section 5.4.3.

The systems in this evaluation differ in the compression stage. The baseline does not use any compression and presents only original source sentences to the sentence selector. The second system uses HMM Hedge to generate multiple compressions of source sentences as described in Section 3.2.3. The third and fourth systems use Trimmer to generate the compressions. As described in Section 4.4, Trimmer can generate multiple compressions by treating the intermediate stages of trimming as candidates, or by using multi-candidate Trimmer rules. The third system, labeled Trimmer, generates multiple compressions by using the intermediate stages of Trimmer processing as candidates, and the fourth system, Trimmer-MCR, uses the three multi-candidate rules described in Section 4.4 to generate the candidates.

The results of this evaluation are shown in Table 5.2. The systems using

	No Compression	HMM Hedge	Trimmer	Trimmer-MCR
R1 Recall	**0.27381** (0.26607-0.28186)	**0.27730** (0.26982-0.28486)	**0.29314** (0.28489-0.30164)	**0.29375** (0.28553-0.30215)
R2 Recall	**0.06144** (0.05782-0.06534)	**0.06525** (0.06152-0.06943)	**0.06554** (0.06198-0.06938)	**0.06575** (0.06202-0.06979)

Table 5.2: Rouge scores and 95% confidence intervals for 50 DUC2006 test topics, comparing four compression approaches within the MuD MASC framework.

Trimmer for sentence compression performed significantly higher than the system using HMM Hedge and the baseline for Rouge-1 but there was not a significant difference among the systems for Rouge-2. The Trimmer system that used multi-candidate rules in the generation of compressed candidates scored non-significantly higher on Rouge-1 and Rouge-2 than the Trimmer system using only intermediate stages of fixed-order rule applications.

MuD MASC using intermediate stage Trimmer for compression was submitted to the official DUC2006 evaluation. Results show that use of sentence compression hurt the system on human evaluation of grammaticality. This is not surprising, since Trimmer aims to produce compressions that are grammatical in Headlinese, rather than standard English. The submitted MuD MASC system output scored significantly lower than 23 systems on NIST's human evaluation of grammaticality. However, the system did not score significantly lower than any other system on NIST's human evaluation of content responsiveness. A second NIST evaluation of content responsiveness asked evaluators to take readability into consideration. In

	Rouge-2	Rouge-SU4	BE-HM
MCR Score	0.0805	0.1360	0.0413
Higher	1	1	0
Not Different	23	24	27
Range	0.0678-0.0899	0.1238-0.1475	0.0318-0.0508
Lower	11	10	8

Table 5.3: Official DUC2006 Automatic Metrics for MuD MASC using intermediate stage Trimmer for sentence compression (System 32).

this evaluation, MuD MASC scored significantly lower than only two systems. The evaluators recognized that Trimmer compressions are not grammatical in standard English; yet, the content coverage was not significantly different from the best automatic systems, and only two systems were found to be significantly more readable.

NIST computed three “official” automatic evaluation metrics for the DUC2006: Rouge-2, Rouge-SU4 and BE-HM. Table 5.3 shows the official scores of the submitted MuD MASC system for these three metrics, along with numbers of systems that scored significantly higher, significantly lower, or were not significantly different from MuD MASC. Also shown is the range of scores for the systems that were not significantly different from MuD MASC. These results show that the performance of MuD MASC was comparable to most other systems submitted to DUC2006.

The evaluation in Table 5.2 suggests that Trimmer sentence compression is preferable to HMM Hedge sentence compression for generation of English summaries of collections of document in English. However, HMM Hedge may prove to have value with noisier data, as discussed in Section 6. Nevertheless, sentence compression appears to be a valuable component of the framework for multi-document summarization, thus validating the ideas behind Multi-Candidate Reduction.

5.5 Sentence Compression as an Interchangeable Component of a Multi-Document Summarization System

The University of Maryland and Institute for Defense Analysis Center for Computing Sciences (IDA/CCS) collaborated on a different submission to DUC2006. Both sites' systems included tools for sentence compression (i.e., *trimming*), sentence selection (i.e., scoring) and summary generation. UMD and IDA/CCS systems were merged in a variety of ways. Some tools could be used in combination; others were treated as alternatives. For example, the sentence compression tools could be used separately, in combination, or omitted entirely. Likewise, sentence scoring could be done by a combination of tools or by each site separately. However, only one or the other of the summary generation tools was used in any given configuration.

For DUC2006, UMD and IDA/CCS submitted the configuration that had the highest Rouge-2 score on the DUC2005 test data.

This section will examine the effect of the different sentence compression approaches on multi-document summarization. Trimmer and CCS sentence trimming differ in their level of risk taking. CCS sentence trimming is conservative with respect to grammaticality; Trimmer over-generates possible compressions and takes more risks with respect to content and structure.

5.5.1 IDA/CCS Sentence Compression

Trimmer and IDA/CCS sentence trimming both use syntactic trimming, but differ in the depth of parsing information used and the level of risk assumed. The CCS trimmer is conservative, using shallow parsing. The Trimmer is aggressive and uses full parses.

The CCS trimmer aims to remove parts of sentences that are less likely to contain information that would be important to have in a summary *without* having an impact on the grammaticality of the sentence. This is achieved by matching each sentence to established patterns that key off of specific words and/or punctuation to locate phrases or clauses that can be deleted.

Removals in the CCS system include:

- lead adverbs, conjunctions, and semantically light multi-word phrases (such as “As an example,” or “At one point.”;
- medial adverbs, such as “also,” “however”;
- age references, as in “, 51” or “, aged 24”;
- gerund phrases;
- relative clause appositives; and
- attributions, such as “police said.”

CCS sentence trimming is described in greater detail in Conroy et al. (2006b).

5.5.2 IDA/CCS Candidate Scoring: Omega Estimated Oracle Score

The CCS method for scoring candidates uses an oracle approximation approach, referred to here as the omega score. A set of human-generated summaries for a topic τ can be used to calculate a maximum-likelihood estimate of the probability that a human would select term t for a summary of τ, $\hat{P}(t|\tau)$. The maximum-likelihood estimate of an oracle score for a sentence x is

$$\hat{\omega}(x) = \frac{1}{|x|} \sum_{t \in T} x(t) \hat{P}(t|\tau)$$

where $|x|$ is the number of distinct terms in x, T is the set of all terms used in τ and $x(t) = 1$ if x contains t, and 0 otherwise.

In the absence of a set of human-generated summaries $P(t|\tau)$ is estimated by assuming that query terms extracted from the topic description and signature terms extracted from the documents in the topic set can stand in for human summaries. Given a set of query terms and signature terms, $P(t|\tau)$ is estimated as

$$P_{qs}(t|\tau) = \frac{1}{2} q_t(\tau) + \frac{1}{2} s_t(\tau)$$

where $s_t(\tau) = 1$ if t is a signature term for τ and 0 otherwise, and $q_t(\tau) = 1$ if t is a query term for τ and 0 otherwise. The estimate is improved by using pseudo-relevance feedback. The top k scoring sentences according to ω_{qs} are taken to be an extractive summary, and $\rho(t)$ is the expectation that t occurs in these sentences. The improved estimate of $P(t|\tau)$ is the average of $P_{qs}(t|\tau)$ and $\rho(t)$:

$$P_{qs\rho}(t|\tau) = \frac{1}{2} P_{qs}(t|\tau) + \frac{1}{2} \rho(t)$$

Thus the omega score for a sentence x is given by

$$\omega_{qs\rho}(x) = \frac{1}{|x|} \sum_{t \in T} x(t) P_{qs\rho}(t|\tau)$$

See Conroy et al. (2006a) and Conroy et al. (2006b) for more details about the omega score.

5.5.3 IDA/CCS Summary Generation

The IDA/CCS system (Conroy and O'Leary, 2001; Conroy et al., 2006a) forms a summary by taking the top scoring candidates among those candidates containing at least 8 distinct terms. (The length of 8 was empirically determined to be optimal using the DUC2005 test data.) Top-scoring sentences are selected to create a summary of length 500, i.e., twice the target length. Redundancy is minimized by using a pivoted QR matrix decomposition to select a subset of these sentences that has the desired length.

The lead sentence of the summary is the highest scoring candidate as given by the score $\omega_{qs\rho}$. The order for the subsequent sentences is determined using a Traveling Salesperson (TSP) formulation that is seeded with the identified lead sentence and the set of subsequent sentences selected by the pivoted QR decomposition. A *distance* between each pair of sentences was defined and an ordering was then determined that minimized the sum of the distances between adjacent sentences. Since the summaries were limited to 250 words, it is not too difficult to solve the TSP. For example, there are only 3,628,800 ways to order 10 sentences plus a lead sentence, so exhaustive search is feasible. Rather than do an exhaustive search, however, the best of a large sample of orderings, some random and some determined by single-swap changes on a previous candidate ordering, was chosen.

5.5.4 System Combinations

A common pre-processing base was developed by IDA/CCS from which to proceed with various combinations of system components. This base consisted of sentence splitting, "junk removal," and the initial sentence selection. The "junk removal" deals with removal of datelines, editor's instructions, and other strings of characters that should not appear in a summary and are not true parts of sentences. The initial sentence selection used the omega score to select twice as much material as would fit in the allowed space, in this case 500 words. The collaborative test systems all used this pre-processed base as the source from which a 250-word summary was generated.

The collaboration enabled experimentation with a variety of options for sentence trimming and sentence selection:

- There were two sentence compression systems, IDA/CCS trimming and Trimmer. They can be used together, independently, or trimming can be omitted, giving four options. When both compression systems are used CCS trimming is applied first, then UMD trimming generates multiple trimmed candidates.[5]

- There were two sentence selection systems, IDA/CCS's CLASSY and UMD's MuD MASC. These systems operate independently, giving two options.

- The summary generation systems make use of features to select high-relevance, low-redundancy sentences. Each system can use URA relevance and centrality

[5] IDA/CCS trimming is applied first for two pragmatic reasons. First, it generates only one trimmed candidate which can serve as a candidate or as the input to Trimmer. Second, IDA/CCS trimming is generally subsumed by Trimmer, so applying IDA/CCS trimming after Trimmer would be difficult to detect.

scores, omega scores or a combination of URA relevance and centrality scores and omega, giving three options for scoring candidates. When omega is used on Trimmer candidates, omega is re-calculated for each candidate.

For the IDA/CCS sentence selection, the trimming method determined how the URA and omega scores were used. If Trimmer was used, the "best candidate trimming" of a sentence was selected by using a weighted combination of the URA and omega scores. For combined systems that used only omega, the weight of URA was set to zero. For combined systems that used only URA, the weight of omega was set to zero. If the omega weight was non-zero, the columns of the term sentence matrix were normalized to the omega score. If the omega weight was zero, the URA score was used to weight the columns.

If Trimmer was not used then there is no need to select a "best candidate trimming," and all sentences were sent to the pivoted QR. The weighting of columns of the term sentence matrix in the QR is handled in the same manner.

The UMD summary generator combined the URA and omega scores as part of the linear combination of features that are used to rescore the candidates at each iteration. When URA features and omega were used together, each of the four URA features was given weight 1.0 and the omega score was given weight 4.0. When only omega was used, the URA features were given weights 0; and when only URA was used omega was given weight 0.

This gives a total of 24 combination systems that were tested on the DUC2005 test data, computing average Rouge-1 and Rouge-2 scores for each combination

CCS Trimmer	Y	Y	Y	Y	Y	Y
UMD Trimmer	Y	Y	Y	Y	Y	Y
Summary Generation	MuD MASC	MuD MASC	MuD MASC	CCS	CCS	CCS
Omega	Y	Y	N	Y	Y	N
URA	Y	N	Y	Y	N	Y
Rouge-1	0.3776	0.3762	0.3819	0.3837	0.3847	0.3846
Rouge-2	0.0770	0.0772	0.0785	0.0776	0.0792	0.0775

Table 5.4: Using Both UMD and CCS Sentence Compression

CCS Trimmer	Y	Y	Y	Y	Y	Y
UMD Trimmer	N	N	N	N	N	N
Summary Generation	MuD MASC	MuD MASC	MuD MASC	CCS	CCS	CCS
Omega	Y	Y	N	Y	Y	N
URA	Y	N	Y	Y	N	Y
Rouge-1	0.3870	0.3877	0.3882	0.3879	0.3881	0.3864
Rouge-2	0.0788	0.0796	0.0793	0.0784	0.0790	0.0780

Table 5.5: Using Only CCS Sentence Compression

CCS Trimmer	N	N	N	N	N	N
UMD Trimmer	Y	Y	Y	Y	Y	Y
Summary Generation	MuD MASC	MuD MASC	MuD MASC	CCS	CCS	CCS
Omega	Y	Y	N	Y	Y	N
URA	Y	N	Y	Y	N	Y
Rouge-1	0.3773	0.3751	0.3817	0.3847	0.3822	0.3863
Rouge-2	0.0765	0.0769	0.0781	0.0769	0.0776	0.0776

Table 5.6: Using Only UMD Sentence Compression

CCS Trimmer	N	N	N	N	N	N
UMD Trimmer	N	N	N	N	N	N
Sentence Sel	UMD	UMD	UMD	CCS	CCS	CCS
Omega	Y	Y	N	Y	Y	N
URA	Y	N	Y	Y	N	Y
Rouge-1	0.3860	0.3881	0.3886	0.3894	0.3872	0.3882
Rouge-2	0.0778	0.0773	0.0783	0.0785	0.0794	0.0781

Table 5.7: Using No Sentence Compression

scored against the human abstracts. The results are described in the next section.

5.5.5 Selection of a System Combination

The Rouge results for each combination are given in Tables 5.4 through 5.7. The ranks of the combination systems on DUC2005 test data are shown in Tables 5.8 and 5.9. At 95% confidence, there are no significant differences among the Rouge-2 scores, and for Rouge-1, the top 7 were significantly different from the lowest two systems. Although the systems using Trimmer compression tended toward the lower half, it is not possible to conclude with certainty that Trimmer compression is helping or harming the summaries with respect to Rouge. The system with the highest Rouge 2 Recall score on the previous year's test data (DUC2005) was selected for submission to DUC2006. This system used both IDA/CCS trimmer and Trimmer, and the IDA/CCS summary generator with both URA and omega to select among the Trimmer sentence variations and omega only to make the final sentence selections, making it one of the most truly collaborative of the possible combinations.

Id	Selector	IDA/CCS Trim	UMD Trimmer	Candidate Score	Rouge-1
1	MuD MASC	no	no	URA	0.38865
2	MuD MASC	yes	no	URA	0.38818
3	CLASSY	yes	no	omega	0.38813
4	CLASSY	yes	no	URA & omega	0.38813
5	CLASSY	yes	no	URA	0.38813
6	MuD MASC	no	no	omega	0.38812
7	MuD MASC	yes	no	omega	0.38775
8	CLASSY	no	no	URA	0.38720
9	CLASSY	no	no	omega	0.38720
10	CLASSY	no	no	URA & omega	0.38720
11	MuD MASC	yes	no	URA & omega	0.38697
12	CLASSY	no	yes	URA	0.38627
13	MuD MASC	no	no	URA & omega	0.38601
14	CLASSY	yes	yes	URA & omega	0.38570
15	CLASSY	no	yes	URA & omega	0.38563
16	CLASSY	yes	yes	omega	0.38474
17	CLASSY	yes	yes	URA	0.38465
18	CLASSY	no	yes	omega	0.38225
19	MuD MASC	yes	yes	URA	0.38193
20	MuD MASC	no	yes	URA	0.38167
21	MuD MASC	yes	yes	URA & omega	0.37759
22	MuD MASC	no	yes	URA & omega	0.37729
23	MuD MASC	yes	yes	omega	0.37619
24	MuD MASC	no	yes	omega	0.37507

Table 5.8: Rouge-1 Average Recall ranking on DUC2005 data set. The systems differ on which summary generation system (candidate selector) was used, whether CCS trimming was used, whether UMD trimming was used, and which combination of URA and omega was used to score candidates. The rows are sorted by Rouge-1.

Id	Selector	IDA/CCS Trim	UMD Trimmer	Candidate Score	ROUGE-2
1	CLASSY	yes	yes	URA & omega	0.07988
2	MuD MASC	yes	no	omega	0.07964
3	CLASSY	no	no	URA	0.07935
4	CLASSY	no	no	omega	0.07935
5	CLASSY	no	no	URA & omega	0.07935
6	MuD MASC	yes	no	URA	0.07930
7	CLASSY	yes	yes	omega	0.07924
8	CLASSY	no	yes	URA & omega	0.07913
9	CLASSY	yes	no	omega	0.07897
10	CLASSY	yes	no	URA & omega	0.07897
11	CLASSY	yes	no	URA	0.07897
12	MuD MASC	yes	no	URA & omega	0.07880
13	MuD MASC	yes	yes	URA	0.07847
14	MuD MASC	no	no	URA	0.07830
15	MuD MASC	no	yes	URA	0.07808
16	MuD MASC	no	no	URA & omega	0.07783
17	CLASSY	no	yes	omega	0.07763
18	CLASSY	no	yes	URA	0.07757
19	CLASSY	yes	yes	URA	0.07747
20	MuD MASC	no	no	omega	0.07730
21	MuD MASC	yes	yes	omega	0.07722
22	MuD MASC	yes	yes	URA & omega	0.07705
23	MuD MASC	no	yes	omega	0.07685
24	MuD MASC	no	yes	URA & omega	0.07649

Table 5.9: ROUGE-2 Average Recall ranking on DUC2005 data set. The systems differ on which summary generation system (candidate selector) was used, whether CCS trimming was used, whether UMD trimming was used, and which combination of URA and omega was used to score candidates. The rows are sorted by ROUGE-2

System	Rouge-1	Rouge-2	Rouge-SU4
UMD–IDA/CCS	0.36026 (0.35407-0.36663) 7th of 35	0.08954 (0.08540-0.09338) 4th of 35	0.14607 (0.14252-0.14943) 4th of 35
CLASSY	0.40279 (0.39649-0.40839) 5th of 35	0.09097 (0.08671-0.09478) 2nd of 35	0.14733 (0.14373-0.15069) 3rd of 35
MuD MASC Trimmer	0.38196 (0.37597-0.38774) 17th of 35	0.08051 (0.07679-0.08411) 13th of 35	0.13600 (0.13212-0.13955) 13th of 35

Table 5.10: Official DUC2006 Rouge average recall scores, 95% confidence intervals and ranks for UMD–IDA/CCS, CLASSY and MuD MASC Trimmer

5.5.6 Evaluation of Combination Systems

In the DUC2006 evaluation, the UMD–IDA/CCS combination system was System 8. The IDA/CCS submission, CLASSY, was System 15 and the Trimmer submission was System 32. Tables 5.10 and 5.11 show the scores and ranks of the three systems. The human responsiveness score, according to the assessor instructions, measures "the amount of information in the summary that actually helps to satisfy the information need expressed in the topic statement." The overall responsiveness score also measure content, but "is based on both the readability of the summary and the amount of information in the summary that helps to satisfy the information need expressed in the topic." Both responsiveness scores are based on a scale of 1 (very poor) to 5 (very good).

Due to its conservative approach, the IDA/CCS trimmer does not introduce many grammatical errors The IDA/CCS trimmer permits the inclusion of at least 2–3 additional sentences in a summary.

System	Responsiveness Content	Responsiveness Overall
UMD–IDA/CCS	2.58 15th of 35	1.96 32nd of 35
CLASSY	2.48 23rd of 35	2.06 27th of 35
MuD MASC Trimmer	2.60 13th of 35	2.08 23rd of 35

Table 5.11: Official DUC2006 Responsiveness Scores for UMD–IDA/CCS, CLASSY and MuD MASC Trimmer

In combination with the Sentence Selector, use of Trimmer adds on average 2.73 sentences to a summary. This represents a net gain over Sentence Selector without Trimmer. On average Trimmer introduces 3.13 new sentences to a summary but drops 0.40 existing sentences from the untrimmed summary. This average is not affected by the use of the IDA/CCS trimmer, however it is affected by the features used in sentence selection. When only URA is used, 1.72 sentences are added by Trimmer, but when only omega is used, 3.95 sentences are added. One might guess that this effect is largely due to omega's bias for shorter sentences, with or without Trimmer. However, this appears not to be the case. The average summary generated without Trimmer using only URA contained 11.2 sentences, while the average summary generated without Trimmer using only omega contained 12.0 sentences. The difference appears to be in how many original source sentences are replaced by a trimmed candidate. With UMD trimming and URA, 51.3% of sentences are replaced by a trimmed version of that sentence. Under omega, 65.4% are replaced by a trimmed version. When both URA and omega are used, the figures fall in between: 54.8% of sentences are replaced by a trimmed candidate, resulting in an average net increase of 2.51 sentences.

Both the IDA/CCS CLASSY and UMD/BBN Trimmer MuD MASC DUC2006 submissions use sentence compression and candidate selection, yet CLASSY scored significantly higher than Trimmer for Rouge-1, Rouge-2 and Rouge-SU4 evaluations. One might interpret this result to mean that the aggressive sentence compressions provided by Trimmer are actually harmful in comparison with the conservative compression used by CLASSY. Yet the collaborative system, which also uses Trimmer sentence compression, is very close to CLASSY for Rouge-2 and Rouge-SU4. Note that the collaborative system and CLASSY share a filtering stage based on omega scores, while Trimmer filtering consists of using the first five sentences of each document. The lack of a significant difference of the collaborative system and CLASSY suggests that the difference between CLASSY and Trimmer is more likely a result of differences in filtering than compression.

It is also important to note that the responsiveness scores for Trimmer and CLASSY are very close, despite the significant differences in the automatic evaluation. To understand this discrepancy, it helps to examine the Rouge evaluation more closely. Figures 5.8 and 5.9 show the system outputs for topic D0601A. The terms in bold are those which actually contributed to the Rouge-2 score, with Rouge configured to ignore stopwords. The individual summary scores show that for this example Rouge gave a much higher score to CLASSY (0.04036 for 18 bigram matches) than to Trimmer (0.02242 for 10 bigram matches), yet it does not appear that the CLASSY summary is more responsive to the query, shown in Figure 5.4, Participation in the Agriculture Department's Food Distribution Program on **Indian Reservations** increased 8.2 percent. ○ David Rocchio, deputy legal counsel to Vermont Gov. Howard Dean, who has been involved in discussions on Indian gambling through the National Governors' Association, said that "the concern that governors have is not with the benefit **casinos bring** to tribes.". ○ **American Indian** reservations would get an infusion of $1.2 billion. ○ Federal programs include the **Native American** Employment and Training Programs, grants to local education agencies for Indian education, and family violence prevention and services. ○ **Native Americans** living on reservations that maintain 50 percent or more **unemployment** are **exempt** from the national five-year family limit on welfare benefits. ○ Approximately 4,000 **Native Americans** sought treatment in the San Francisco area, according to the Indian Health Service, a branch of the U.S. Department of Health and Human Services. ○ Members have a right to **live** on the **reservation**, but many survive the consequence, according to Robert Coulter, a lawyer with the Indian Law Resource Center, a nonprofit group. ○ Juvenile crime is one strand in the web of social **problems facing** urban and reservation Indian communities. ○ The main purpose of the visit – the first to a reservation by a president since Franklin Roosevelt – was simply to pay attention to **American Indians**. ○ Smith and thousands are seeking help for their substance abuse at the **American Indian** Community House, the largest of a handful of **Native American** cultural institutions in the New York area. ○ Soldierwolf's family represents the problems that plague ○

Figure 5.8: Trimmer submission for DUC 2006 topic D0601 with terms that contribute to the Rouge-2 score in bold.

As **crime rates** fall nationwide, they are rising in **American Indian** communities, especially among the 43 percent of Indians under age 20. ◦ Juvenile crime is one strand in the web of social **problems facing** urban and reservation Indian communities. ◦ **Indian tribes** across the country have taken various steps to curb **alcohol-related** problems. ◦ The Shinnecock are a state-recognized tribe, but one of more than 100 **Indian tribes** not acknowledged by the **federal government**. ◦ Federal lawmakers soon must sort out the muddled state of **American Indian** gambling. ◦ An Indian who commits a crime outside a reservation is destined for the usual **federal, state** or local courts. ◦ Poverty among the Sioux on the Pine Ridge **Indian Reservation** in South Dakota makes theirs the poorest county in the country. ◦ The poverty rate in Kings County, which includes the tribe's small reservation, climbed from 18.2 percent in 1989 to 22.3 percent in 1995. ◦ Over all, the rate of substance abuse among **Native American** adults is over 20 percent nationwide. ◦ An estimated 50 percent of **American Indians** are unemployed, and at Pine Ridge the problem is even more chronic – 73 percent of the people do not have jobs. ◦ Indians account for less than 1 percent of the U.S. population today, spread mostly across the West on 314 reservations. ◦ According to statistics from the Census Bureau and the **Bureau** of **Indian Affairs**, there are **1.43 million Indians living** on or near **reservations**. ◦ The **Bureau** of **Indian Affairs** estimates that at least half the **American Indian population**...

Figure 5.9: CLASSY submission for DUC 2006 topic D0601 with terms that contribute to the Rouge-2 score in bold.

than Trimmer. Nor does it appear that either system has broader coverage: both mention gambling, crime, substance abuse, and unemployment. CLASSY additionally covers poverty and federal recognition, which are not mentioned by Trimmer, but Trimmer mentions federal assistance and health issues which are not covered by CLASSY. This is a typical example. Although CLASSY does a better job on average of matching n-grams from the reference summaries, it does not appear that the CLASSY summaries are actually more responsive or cover more information than the Trimmer MuD MASC summaries.

CLASSY's omega score, which uses query terms and signature terms derived from the documents to simulate an oracle, has a strong bias for sentences rich in topic terms. Trimmer MuD MASC uses URA query relevance and document centrality, which correspond in function to query and signature terms, as well as a variety of compression and length features. The use of other features tempers MuD MASC's bias for topic-term rich candidates with a bias for concise sentences, allowing it to create summaries with high responsiveness and high information coverage.

If the difference in candidate scoring accounts for the difference in Rouge scores, there should be a measurable difference in the term frequency, tf, scores of the matching n-grams that are the overlap between the peer summaries and the model summary references.[6] An investigation of the matching unigrams shows that there is a difference in average matching unigram tf scores between CLASSY and MuD MASC. For each document set in the DUC2006 test data, three sets of match-

[6] Because these are multi-document summaries, the documents in the *idf* calculation are really sets of documents. Given the small number of document sets in the corpus, it is not clear that *idf* is an applicable measure for this situation.

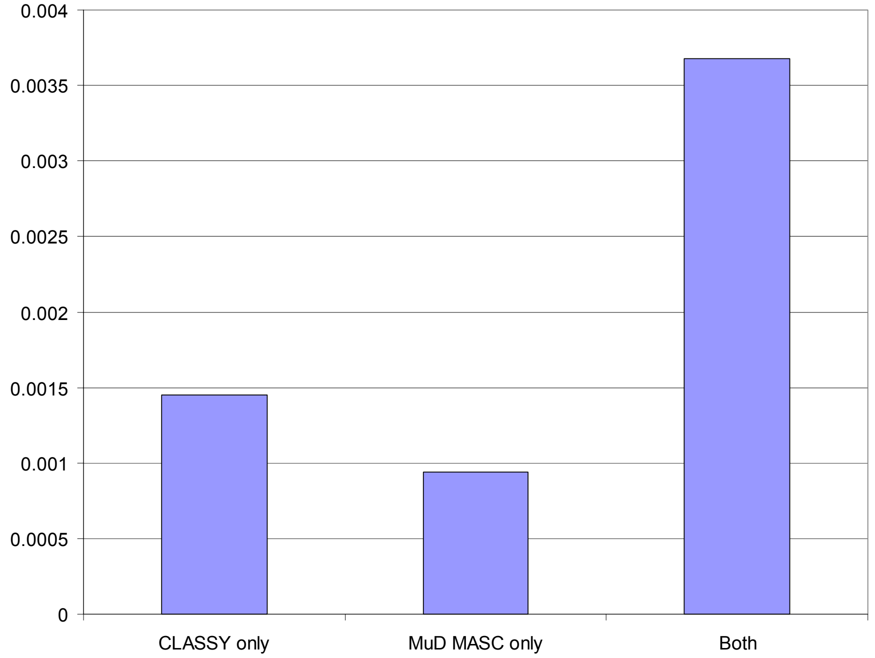


Figure 5.10: Average term frequencies of matching unigrams for CLASSY and Mud MASC.

ing unigrams were found: those matched by both CLASSY and MuD MASC, those matched only by CLASSY, and those matched only by MuD MASC. Figure 5.10 shows the average term frequencies of these unigrams. The highest term frequency matching unigrams are those matched by both systems. However, the unigrams matched by CLASSY only have a substantially higher average term frequency than those matched only by MuD MASC. CLASSY's greater ability to bias its candidate selection in favor of high term frequency words accounts for its higher Rouge scores.

5.6 Summary

In this chapter the tools developed for single document summarization were adapted to the task of multi-document summarization. A human study demonstrated that the use of sentence compression in multi-document summarization can potentially yield a 17% reduction in size without loss of relevance. Multi-document summarization systems were presented that used HMM Hedge and Trimmer as the source of sentence compression, as well as a baseline that did not use sentence compression. An automatic evaluation showed the promise of using sentence compression under the MASC framework to improve system performance over the baseline, although the improvements in Rouge-2 did not reach the level of significance.

A collaboration between University of Maryland and IDA/CCS resulted in choosing a system that used two kinds of sentence compression, conservative and aggressive, because this was the combination that scored highest on the previous year's test set. An analysis of the system output revealed that use of Trimmer sentence compression in multi-document summarization did make room for additional sentences in the summary. A closer examination of Trimmer and CLASSY system output from the DUC2006 evaluation shows that, even though CLASSY performed significantly higher on the automatic evaluation, there was little difference between the systems in the level of overall content responsiveness.

Chapter 6

Novel Genres and Applications

HMM Hedge, Trimmer and Topiary were developed in the domain of written news for summarization tasks involving monolingual, formal text. This chapter describes the application of sentence compression to two novel genres (broadcast news transcript and email thread summarization) and two new applications (cross-language information retrieval (CLIR) and structured queries). Issues affecting the applicability of Trimmer and Topiary to these new genres and applications will be discussed in detail.

6.1 Broadcast News Summarization

Trimmer and Topiary were applied to the task of creating 75-character summaries of single broadcast news stories. The corpus for this task consisted of 560 broadcast news stories. The sources include ABC, CNN, NBC, Public Radio International and Voice of America. All stories were broadcast during January 1998 to June 1998 or October 2000 to December 2000. The stories were transcribed from the audio by BBN's BYBLOS Large Vocabulary Continuous Speech Recognition (Colthurst et al., 2000). An example from the test corpus is shown in Figure 6.1. The Document boundary detection was also provided by BBN.

> For his part Fidel Castro was the ultimate political Survivor. People have predicted his demise so many times and the us has tried to hasten it on several occasions. Time and again he endures. He has outlasted and sometimes outsmarted eight American presidents. Fidel Castro invited John Paul to come for a reason. This is clearly an opportunity for Cuba to look good internationally. The entire world will see images of the Pope in Cuba. They'll see images of the Pope with Fidel Castro. ... At the end of the broadcast this evening one more trip around Havana to see what it's been like since the last time and in just a moment Diane Sawyer will have some other news.

Figure 6.1: Excerpt from ABC news story from January 20, 1998, transcribed by BYBLOS

6.1.1 Broadcast News Genre Characteristics

The corpus of broadcast news stories in this test differs from the DUC2003 test corpus in that the DUC2003 documents were the result of IR searches on topics while the broadcast news corpus consists of stories selected from a particular time window. The time window leads to some clustering of stories because certain stories were high-profile news stories during those months. There were 9 documents about the Clinton/Lewinsky scandal from 1998 and 11 on the Supreme Court's involvement in the 2000 presidential election. Many documents were the only document on their topic in the corpus. In contrast the DUC2003 corpus consisted of 60 clusters of 9 to 14 documents on the same topic. The absence of annotated clusters removes the usefulness of the document to cluster centrality score, and the lack of clusters reduces the quality of Unsupervised Topic Detection.

Follow-on work should be done to study broadcast news summarization with a corpus based on Information Retrieval searches, and to develop tools for handling corpora, like this one, based on time windows rather than topic.

Broadcast news has sources of noise that do not apply to written news stories. Preprocessing for these documents includes automatic transcription, document boundary detection and sentence boundary detection. Each step in this process can introduce errors.

There are characteristics of the genre that affect automatic summarization. Broadcast news stories frequently start with short sentences in which speakers identify themselves, their organization, or the specific news program. Broadcast news programs inherently have a temporal narrative flow, unlike newspapers in which distinct stories can be read independently. Broadcast news document frequently includes references to the position of the story in the broadcast (first or last), and transfers from an anchorperson to a field reporter. Follow-on work could detect such structural information to avoid including it in a summary.

6.1.2 Broadcast News Processing

Summaries for the broadcast news corpus were created using Multiple Candidate Sentence Compression with Trimmer and Topiary as described in Sections 4.4.1. The documents were preprocessed by finding sentence boundaries. Two types of sentence filtering were used: either the first non-trivial sentence or the first five non-trivial sentences were considered. Sentences with five or fewer words were considered to be trivial, for example, “Hello, I’m Dan Rather.” BBN’s IdentiFinder was used to tag named entities and time expressions, and Charniak’s parser was used to parse the sentences. Trimmer with multi-candidate rules, described in Section 4.4.1, was used to generate multiple candidate compressions for the sentences. URA centrality scores, described in Section 5.3.1 were calculated for each sentence. URA calculated centralities of sentences to documents and centralities of documents to clusters. There were no explicit clusters in the broadcast news corpus, so the document centralities were not used as a feature. Rouge was used to calculate a score for each candidate, with three human-written 75-character model summaries for each document. BBN's Optimizer was used to set weights for the features enumerated in Section 4.4.1 to optimize Rouge-1 recall.

Optimization was done with different combinations of three groups of features: length-based features (L), rule-based features (R) and centrality features (C), as described in Section 4.4.1, each combination giving a set of feature weights. After optimization, each set of weights was used to select the high-scoring candidate for each document on the linear combination of features. In addition, a simple longest-under-limit baseline was used to select a candidate. Each such system was evaluated with Rouge-1 recall.

Topiary was applied by combining the Trimmer candidates with UTD topics as described in Section 4.4.1.

6.1.3 Examples of Broadcast News Summarization Output

Trimmer is capable of producing trimmed candidates for sentences from broadcast news. The following sentences show source sentences from broadcast news documents, and a compressed candidate proposed by Trimmer.

(44) For the second time in a week a widely recognized American figure has died when he was skiing and ran into a tree.
widely recognized American figure died when he was skiing and ran into tree

(45) Police say a family in upstate New York held captive a mentally impaired mother and daughter and lived off their disability checks for possibly ten years.
family held captive mentally impaired mother and daughter

(46) airlines must tell consumers who ask for the lowest fares that the best deals may be available on their internet sites.
airlines must tell consumers who ask for lowest fares best deals available

The following examples illustrate how Topiary topics can provide minimal summary information when the Sentence Selector makes a poor selection of lead sentence.

(47) (i) Source: all americans are ready to pause for the thanksgiving holiday. but there has been no pause in major events in the race for president.
Trimmer: all americans are ready to pause for the thanksgiving holiday.
Topiary: court recount ballots florida counties: all americans are ready to pause.

(ii) Source: Welcome to this hour of voa news now. I'm Erin brumett in Washington. The incredibly close Florida vote count for President is in the state courts and heading for the us Supreme court at the end of the week.
Trimmer: Welcome to this hour of voa news now.
Topiary: florida supreme court law: Welcome to this hour of voa news now.

6.1.4 Automatic Evaluation of Broadcast News Summarization Approaches

An automatic evaluation using Rouge was performed on the output from the system runs described in Section 6.1.2. The Trimmer variants differ in the number of sentences from the front of the document selected for compression during the filtering stage and the sets of features used during the candidate selection stage. The results

System	First 5 Sentences	First Sentences
Trimmer L	**0.20004** (0.18645-0.21461)	**0.23982** (0.22358-0.25680)
Trimmer R	**0.18989** (0.17459-0.20615)	**0.22946** (0.21271-0.24607)
Trimmer C	**0.06339** (0.05437-0.07288)	**0.13859** (0.12375-0.15294)
Trimmer LR	**0.20301** (0.18899-0.21803)	**0.24481** (0.22782-0.26254)
Trimmer LC	**0.19238** (0.17913-0.20717)	**0.23728** (0.22077-0.25413)
Trimmer RC	**0.19406** (0.17936-0.21017)	**0.22470** (0.20809-0.24109)
Trimmer LRC	**0.20046** (0.18698-0.21603)	**0.24578** (0.22925-0.26305)
Trimmer baseline	**0.17849** (0.16391-0.19449)	**0.21177** (0.19568-0.22744)

Table 6.1: Rouge-1 recall scores for broadcast news single document summarization using Trimmer. Systems differ on how many sentences from the start of the document were selected for compression, five or one, and which sets of features were used in candidate selection. Trimmer baseline used the longest candidate under the length limit.

of the Rouge-1 evaluation on the Trimmer variants are shown in Table 6.1. This evaluation shows that in the filtering stage, selecting only the first sentence of each document is better than selecting the first five sentences. This result is interesting because it shows the same tendency to prefer sentences early in a document that was observed in Section 5.2, despite the differences in genre. The evaluation also shows that better results are achieved by using all three sets of features: length, rule application counts, and centrality scores.

A second evaluation compared systems that varied in the type of compression used: Trimmer, Topiary and a baseline that did not use compression. All

System	Rouge-1	Rouge-2
Baseline	**0.24275** (0.22549-0.25952)	**0.08240** (0.07058-0.09458)
Trimmer	**0.24578** (0.22925-0.26305)	**0.08404** (0.07334-0.09491)
Topiary	**0.25460** (0.23840-0.27153)	**0.08411** (0.07366-0.09546)

Table 6.2: Rouge-1 and Rouge-2 recall scores for broadcast news single document summarization using first-sentences. Systems differ in the type of sentence compression used for candidate generation. The baseline used no sentence compression. Trimmer and Topiary, with multi-candidate rules, were also used to generate candidates.

systems selected only the first sentence in the filtering phase. The Trimmer system used multi-candidate rules and selected among the candidates based on length, rule-count, and centrality features. The Topiary system also used multi-candidate rules and selected using length, rule-count, centrality, and topic features. The results of this evaluation are shown in Table 6.2. This evaluation shows that use of sentence compression has potential to improve performance of summarization in the domain of broadcast news, although the improvement has not yet reached the level of statistical significance.

Despite the difference in style between broadcast news and written news, it is still the case that always choosing the first non-trivial sentence as the lead scores higher on Rouge-1 recall than approaches that look farther into the document, using current tools. In Section 4.4.1 it was observed that the best scores on Rouge-1 recall for Trimmer summaries of written news were achieved by optimizing on length, rule and centrality features together. The pattern holds for broadcast news, as the system trained on L, R and C features has the highest Rouge-1 score. In the second evaluation, systems using sentence compression score higher than a baseline that does not use any compression. Finally, the Rouge-1 score for the Topiary system was higher than the best-scoring Trimmer system. The trends observed in this evaluation mirror the observations about Trimmer and Topiary applied to written news, and demonstrate the application of compression-based summarization approaches to a novel genre.

6.2 Email Thread Summarization

The problem of summarizing email threads is technically challenging because email is qualitatively different from newswire text. Unlike single-author journalistic writings, email threads capture the conversation among two or more individuals, across both time and space. Another difference is that newswire text is meant for general consumption by a wide audience, while emails are only intended for their recipients. As a result, emails are much more informal and often rely on shared contexts, specialized sublanguages, and other implicit cues to facilitate efficient communication. Furthermore, email is often embedded in a larger organizational context which cannot be directly observed from the texts alone, as in the simple case of collaboration between two colleagues that occurs partially over email and partially in face-to-face meetings. Finally, email is not subjected to the careful editorial process that news articles are, thus making typos, incomplete sentences, and other grammatical oddities much more prevalent.

Email represents an instance of “informal” text—a broader genre that includes conversational speech, blogs, instant and SMS messages, etc. Interest in automated processing techniques for informal media has been growing over the past few years for many reasons. There is the recognition that an increasingly large portion of society's knowledge is captured in informal communication channels. Serious research in this area is facilitated by the availability of large collections and the falling cost of computational and storage resources. Finally, informal media push the frontiers of human language technologies by forcing researchers to develop more general and robust algorithms that are adaptable to different domains and tasks.

An email thread is a collection of messages that form a multi-party conversation. Generally, a thread will consist of an initial email message and subsequent responses to it. As a first step, the MASC summarization framework has been adapted to tackle this problem. This first foray paves the way for future advances in the area.

In addition to the problem of generating content, there are also several presentational issues associated with email thread summarization. The usual practice of presenting an undifferentiated segment of prose does not appear to be a good idea, since email comes with a great deal of metadata (e.g., sender, recipients, time, etc.). Presentational issues potentially confound evaluations of content since associated metadata may be required for the interpretation of system output.

Finally, evaluation issues in general present challenges. Are established methodologies for existing summarization tasks applicable? Do automatic metrics such as Rouge predict human judgments? If not, are there other alternatives? Despite these open research questions, Rouge will be used to evaluate the email summarization systems presented here because of its acceptance in the DUC summarization evaluations. In this specific case, evaluation is rendered more complex by the highly technical domain.

6.2.1 Email Summarization Framework

This section will present two different applications of the MASC framework to the problem of email thread summarization. In one case, each message can be considered a “document” in a multi-document summarization task. In the same way that traditional systems are given a number of documents about a topic and asked to generate a summary, this approach treats each email as a document “about” the topic. This approached is called the Collective Message Summarization (CMS) approach. In contrast, an alternative view treats email thread summarization as the task of generating successive single-document summaries. That is, a short summary is generated for each individual email, and then the output is aggregated to form a complete summary. This approach is called Individual Message Summarization (IMS).

Both approaches have advantages and disadvantages. While IMS will faithfully preserve thread structure, it is fairly obvious that not all messages in a thread are equally important. Thus, the approach runs the risk of over-representing messages that do not contain important content. Furthermore, since summary length is largely determined by thread length, system output must be further processed to generate a summary of a given length. The CMS approach has the opposite problems: although

Human	Avg. Size (words)
1	127.4
2	53.2
3	136.7
4	137.6
5	242.5

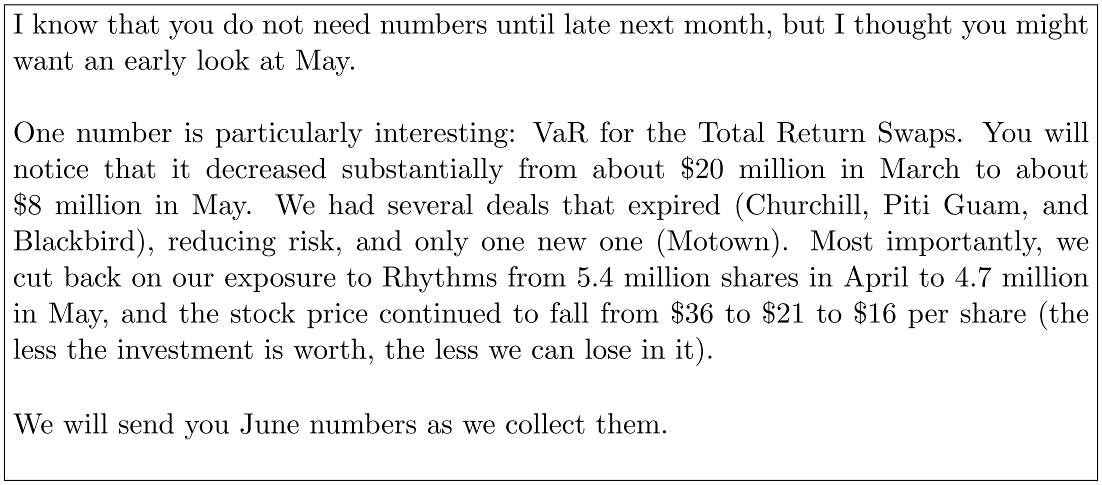
Table 6.3: Average size in words of the model summaries for the email threads

summary length is easier to control, it is more difficult to convey thread structure (and hence the conversational nature of email). There is little guarantee that content in different parts of the thread will be represented (but this may not be a problem).

6.2.2 The Data

The dataset for this exploration of the email thread summarization problem is the Enron dataset, which consists of approximately half a million emails from the folders of 151 Enron employees. This corpus represents the largest available collection of real-world email traffic, and offers researchers a unique glimpse into the nature of corporate communication and the illegal activities that eventually led to the downfall of the company. Already, many topics have been explored using this data, including name reference resolution (Diehl et al., 2006), topic and role discovery (McCallum et al., 2005), and social network analysis (Diesner et al., 2005). However this work represents the first attempt at summarization on this collection.

Since there were no existing resources to support a summarization task, it was necessary to create a test collection. This was performed by a Master's student



I know that you do not need numbers until late next month, but I thought you might want an early look at May.

One number is particularly interesting: VaR for the Total Return Swaps. You will notice that it decreased substantially from about $20 million in March to about $8 million in May. We had several deals that expired (Churchill, Piti Guam, and Blackbird), reducing risk, and only one new one (Motown). Most importantly, we cut back on our exposure to Rhythms from 5.4 million shares in April to 4.7 million in May, and the stock price continued to fall from $36 to $21 to $16 per share (the less the investment is worth, the less we can lose in it).

We will send you June numbers as we collect them.

Figure 6.2: Text of an email from Thread 6.

in library and information science (LIS) who spent several months learning about energy trading and examining the data (as part of a larger project on knowledge discovery). The test corpus was created with the end application in mind: she first developed information needs that users might have. Using a baseline retrieval engine built on Lucene, she manually searched for relevant threads and selected them for summarization.

In total, ten threads were selected for inclusion in the test collection. The threads range in size from 3 to 30 emails, with an average size of 12.6 emails per thread. In addition to writing a reference summary for each of the threads herself, the Enron expert recruited and trained four additional individuals (also Master's students in LIS) to generate reference summaries. Since these additional subjects had no prior domain knowledge, sessions began with an overview of energy trading and other background necessary to understand the content of the threads (which took a few hours). No length limit was placed on these human reference summaries.

In the end, five reference summaries for each of the ten manually-selected threads were obtained. Table 6.3 shows the average lengths in words of the references, and illustrates the broad range of performance of this task by humans. Summarizer 5 was the Enron expert who assembled the threads and trained the other subjects. She wrote the longest summaries and had the greatest understanding of the domain.

Consider the sample email in Figure 6.2, selected from thread 6. It is apparent that the email thread summarization task on this dataset is very difficult, even for humans. It is obvious that one must be familiar with the arcane world of energy trading in order to comprehend the message contents. Furthermore, this specialized and highly technical domain uses plenty of jargon that is not typically found in newswire text.

All email messages were pre-processed before they were presented to the summarization systems. These processes included removal of headers and attachments. Repetitions of text from earlier messages ("quoted text") was also eliminated. The summarization systems were presented with the cleanest possible text.

6.2.3 Email Thread Summarization Evaluation

The experiments described in this section explore the problem of email thread summarization. The system task attempted here was to generate one hundred word summaries of threads.

In particular, the experiments focus on two variables:

- Approach: IMS vs. CMS
- Compression method: Trimmer vs. HMM

In the IMS approach, the filtering stage consists of choosing the first non-trivial sentence from each email, where the first non-trivial sentence is the first sentence that is not a salutation or a content-free opening line. After the generation of compressed candidates, the system chooses the best candidate under 75 characters. The character limit was adopted from previous single-document summarization task definitions. In the CMS approach, the sentence selector had access to text in the entire email thread.

Summaries generated by the IMS approach require one additional processing step. Since the length of summaries is determined by the size of the thread, IMS simply retains the first 100 words if the system output is longer than the desired length. Note that additional truncation is not necessary with the CMS approach since summary length is directly controlled by the sentence selector, which iteratively chooses candidates until the desired length has been achieved.

The systems were tested against the following baseline: the first 75 characters of each email message are selected to form a summary. This essentially represents an IMS approach, except without any sentence compression. Since the length of this baseline output is also dependent on thread size, the baseline discards all but the first 100 words.

System output was automatically evaluated using Rouge with the five reference summaries described in the previous section. Table 6.4 shows Rouge-2 recall

Run	Rouge-2
IMS Trimmer	0.0421
IMS HMM	0.0315
CMS Trimmer	0.0453
CMS HMM	0.0508
baseline	0.0489
Human 1	0.0770
Human 2	0.0187
Human 3	0.0963
Human 4	0.0709
Human 5	0.0963

Table 6.4: Rouge recall scores using jackknifing from different system runs.

scores, with jackknifing. Note that since none of the threads were used in system development, they can be considered blind held-out test data. For the sentence selector, the systems employed default parameters trained on data from previous DUC evaluations. In addition, Table 6.4 shows the performance of the human summarizers in order to quantify potential upper-bound performance. For fair comparison, human summaries were also truncated to 100 words. Figure 6.3 offers a different view of the results, with the different conditions sorted in increasing order of Rouge-2 scores. Error bars denote the 95% confidence intervals. This evaluation demonstrates the wide range of human performance on this task, that a simple baseline performs relatively well, and that three of the compression-based systems are statistically equivalent to that baseline.

For reference, sample output from the CMS approach for thread 6 is shown in Figure 6.4—Trimmer output on top and HMM on the bottom. IMS system output for the same thread appears in Figure 6.5. Following Rambow et al. (2004), system

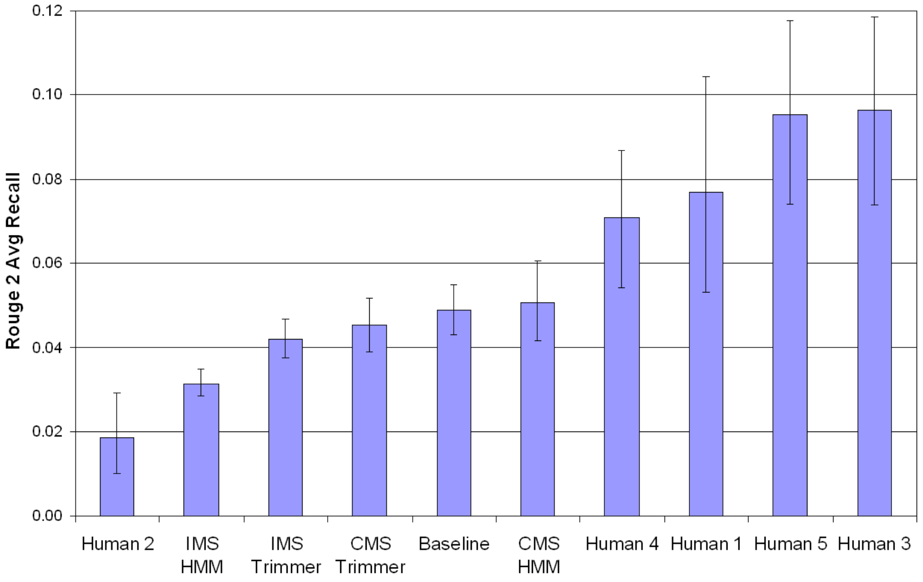


Figure 6.3: Rouge-2 scores for different conditions, sorted in increasing order.

output is sorted chronologically. The author name and a time stamp are prepended to each email. Since sentence breaks are often not explicitly marked, a special break symbol (○) is used for clarity.

The metadata appear purely for the purpose of human readability and are not included in the Rouge evaluations. Although the system output may be difficult to understand, the source text is just as difficult to comprehend due to the prevalence of domain jargon (see Figure 6.2).

6.2.4 Analysis

Two important observations can be made from these initial experiments in email thread summarization: that the task is exceedingly difficult and that the baseline seems to perform well, at least in terms of Rouge scores.

◦ **Eugenio Perez (6/26/2000 06:40):** I know that you do not need numbers until late next month but I thought you might want an early look at May
◦ **Eugenio Perez (10/27/2000 02:50):** The good news was that September VaRs is little changed from the June numbers
◦ **Eugenio Perez (1/25/2001 09:34):** Gary and Georgeanne let me know that all but 487 shares of EOG are hedged (without the EOG leg the Cerberus total return swap is really only a loan and its VaR is about $ 500 thousand)
◦ **Eugenio Perez (2/2/2001 02:14):** AA informed me that the hedges on the New Power Company warrants that were monetized in the Hawaii 125 0 McGarret swaps were put on October 4 not in September

◦ **Eugenio Perez (6/26/2000 06:40):** you might want early look ◦ it decreased substantially ◦ the investment is worth
◦ **Eugenio Perez (10/27/2000 02:50):** New Power Company went public ◦ warrants we inserted are hugely. ◦ swaps will probably be over $30 million.
◦ **Eugenio Perez (1/25/2001 09:34):** VaR fell and $18 million ◦ Cerberus total return swap is really only a loan ◦ natural gas prices are up so much ◦ we can potentially lose
◦ **Eugenio Perez (1/31/2001 08:25):** Please disregard previous versions.
◦ **Eugenio Perez (2/2/2001 02:14):** hedges that monetized 125-0 McGarret swaps put
◦ **Eugenio Perez (2/6/2001 02:20):** we created by granting options ◦ we have long term contracts to remove variability of revenues ◦ the contracts expire ◦ for total return swaps fell from $34 to $28 million.
◦ **Adarsh Vakharia (2/8/2001 09:37):** it is little hedged by Phantom swap ◦ Regards, Adarsh and Eugenio

Figure 6.4: Output from the CMS approach: using Trimmer (top) and HMM-based (bottom).

Summarization of email threads from the Enron dataset is very challenging, even for humans. The primary difficulty comes from the need for specialized domain knowledge in order to comprehend the email messages. Recall that to generate the reference summaries, the domain expert (Human 5) recruited and trained four other subjects for the task. These training sessions, which lasted a few hours, may not have been sufficient. For example, subject 2 found the task so difficult that one of her summaries was simply the following statement: "This thread is very hard

○ **Eugenio Perez (6/26/2000 06:40):** I know that you do not need numbers
○ **Eugenio Perez (10/27/2000 02:50):** September VaRs are little changed from the June numbers
○ **Eugenio Perez (1/25/2001 09:34):** Gary and Georgeanne let me know.
○ **Eugenio Perez (1/31/2001 05:17):** Merchant Assets have recalculated their VaR
○ **Eugenio Perez (1/31/2001 08:25):** Merchant Assets has re-revised numbers.
○ **Eugenio Perez (2/2/2001 02:14):** hedges that monetized 125-0 McGarret swaps put
○ **Eugenio Perez (2/6/2001 02:20):** we entered two swaps to hedge out exposure.
○ **Ardash Vakharia (2/8/2001 09:37):** Georgeanne told Eugenio.
○ **Eugenio Perez (2/9/2001 03:04):** Jedi swap VaR moved.

○ **Eugenio Perez (6/26/2000 06:40):** I did not until late next month I thought you want an early look at
○ **Eugenio Perez (10/27/2000 02:50):** good news VaRs little changed
○ **Eugenio Perez (1/25/2001 09:34):** Gary Georgeanne let me that all but EOG EOG Cerberus swap VaR
○ **Eugenio Perez (1/31/2001 05:17):** Merchant recalculated VaR I have
○ **Eugenio Perez (1/31/2001 08:25):** Merchant Assets has re revised numbers
○ **Eugenio Perez (2/2/2001 02:14):** hedges on New Power Company monetized McGarret swaps
○ **Eugenio Perez (2/6/2001 02:20):** two swaps Enron to hedge
○ **Adarsh Vakharia (2/8/2001 09:37):** Georgeanne tell Eugenio about another stock swap
○ **Eugenio Perez (2/9/2001 03:04):** swap VaR moved from securities trading

Figure 6.5: Output from the IMS approach: using Trimmer (top) and HMM-based (bottom).

to follow. Not sure what they are attempting to convey.” This was reflected in the Rouge score, which was significantly lower than the automatic systems’ (see Figure 6.3).

There is significant variance in human performance on this task. Furthermore, it unclear that humans perform better than machines in terms of Rouge scores. Only 2 of 5 humans scored significantly higher on Rouge-2 recall than the best automated system, and one human performed significantly lower (Subject 2).

The second major observation is the the baseline is highly competitive in terms of Rouge-2 scores, beating all system variants except for CMS HMM (although many of the differences are not statistically significant). Similar baselines have been tough to beat in previous DUC evaluations. In some cases, systems did not perform better than simple baselines until a few years after researchers started tackling the problem (Over and Liggett, 2002). All things considered, system performance is encouraging in this first attempt at email thread summarization on the Enron corpus.

Note that the baseline is essentially a variant of IMS that does not utilize sentence compression, and that the baseline outperforms both IMS HMM and IMS Trimmer. This finding suggests that the sentence compression algorithms are not functioning as expected. However, since results in summarizing newswire data have demonstrated the value of sentence compression (Blair-Goldensohn et al., 2004; Conroy et al., 2006b), out-of-genre issues are likely the culprit. For Trimmer, proper compression depends on correct parse trees, and parsers trained on newswire text (like the one used here) are likely to make many errors. Similarly, language models for HMM were induced from newswire text, which obviously has different distributional characteristics. Using ill-adapted compression techniques appears to be a liability in this particular application.

Nevertheless, it does appear that CMS represents a better approach to email thread summarization than IMS. The CMS HMM variant outperforms the baseline, although differences are not statistically significant. Overall, the CMS performance is encouraging, because the HMM variant performed better than its IMS counterpart, and the same for the Trimmer variant. In the first case, the difference was statistically significant, but not so in the second case.

It is also interesting that Trimmer does not perform significantly better than HMM in either CMS or IMS approaches for the task, even though it was demonstrated that Trimmer performs better than HMM for summarization of written news in multi-document summarization in Section 5.4.4. HMM-based techniques might be a more attractive choice for sentence compression in noisy environments where parser performance is compromised. However, based on these experiments, the HMM-based technique fared worse on out-of-genre text in the IMS case. Statistical methods may not be as robust as previously thought, given that they still rely on language models to capture fluency. Since many *n*-grams in the Enron collection are simply not observed on newswire training data, these language models may not be portable.

Recall that with the IMS approach, only the first 100 words of system output were retained. For longer threads, this resulted in summaries that only covered email messages toward the beginning. This might be problematic, since messages toward the end are expected to contain important information also. For example, the final messages in a thread might discuss the resolution of a particular issue. To test this hypothesis, truncation to 100 words from the end of the system output, for both the IMS and baseline cases, was tried. Unfortunately, results were inconclusive, as Rouge scores remained essentially unchanged.

6.3 Cross-Language Summarization

Cross-language summarization is the generation of a summary in a language that is different from the language (or languages) of the source documents. A typical use for cross-language summarization is in the context of cross-language information retrieval (CLIR), in which users make relevance judgment, answer questions, or perform other tasks using documents in languages they cannot understand. This section will describe how HMM Hedge and Trimmer were adapted to the cross-language problem and describe two experiments in which cross-language summarization was used in a CLIR context. The test language pairs are Hindi-English and Spanish-English.

6.3.1 Cross-Language HMM Hedge

HMM Hedge was ported to headline generation of Hindi stories by using the existing mechanism for morphological variation for verbs, described in Section 3.2.2. In the cross-language application each Hindi story word can generate some set of English headline words. This is handled by creating multiple H states (for English headline words) with corresponding G states (for Hindi story words). For instance, the Hindi word *sa.nvidhaana* can be translated to English as *constitution* or *constituent*. Thus, there will be two H states capable of emitting *sa.nvidhaana*, corresponding to headline words *constitution* and *constituent*

The story language is represented by a unigram model calculated from a corpus of 2976 Hindi stories from the BBC corpus. The stories were translated from the original Devanagari into ITRANS.[1] The stories contain 1,184,603 words from a vocabulary of 56,369 distinct words.

The English translations of Hindi words were taken from a scored Hindi-English lexicon produced by ISI during the surprise-language exercise. The scores indicate the probability that a specific Hindi word can be translated to a specific English word. These probabilities are used in place of the probabilities that a specific story word verb emits a morphological variant in the Headline.

6.3.2 Cross-Language Trimmer

At present, Trimmer is applied to the problem of cross-language headline generation by translating the first sentence of a story into English and running the Trimmer process on the resulting translation. The obvious drawback is that it requires a translation process.

6.3.3 Cross-Language UTD

To find topics for the Hindi stories UTD was run on a corpus of 1M words of Hindi stories from the BBC Hindi corpus. (A native speaker provided a list of stop words.) This produced a set of Hindi topics. Then, the native speaker examined the topics produced and rejected the proposed topics that could not serve as topics (e.g. particles, conjunctions and numbers), and provided English translations of the valid UTD topics. OnTopic was used to assign topics to the test documents.

[1] Devanagari is a script used to write several Indian languages, including Hindi. ITRANS is an ascii character transliteration of Hindi which is readable by native Hindi speakers.

System	Average Headline Length (in words)
HMM1	8.36
HMM2	11.12
Trimmer	12.36
UTD	12.68

Table 6.5: Average Headline Lengths

6.3.4 Hindi to English Summarization for CLIR

HMM Hedge and Trimmer and UTD were applied to the DARPA TIDES-2003 Surprise Language Exercise for Hindi to generate headlines in English for news stories in Hindi. Rouge and a manual evaluation based on human judgments of clarity were applied to HMM Hedge, Trimmer, and UTD cross-language headlines. All evaluation methods make use of four human-generated reference headlines for 25 stories provided by NIST in the Hindi Surprise Language Exercise. Two variants of HMM Hedge were used, HMM1 and HMM2. The difference between HMM1 and HMM2 is that HMM2 was biased to favor longer headlines. The average length of the headlines produced by the systems is shown in Table 6.5. UTD produced an average of 9.10 topics per document, however many of the topics are multi-word topics, and the average number of words was 12.68. Trimmer was run with a length threshold of 15 to produce headlines with average length 12.36.

Sentences (48) through (51) are examples of the system outputs for one document from the evaluated systems.

(48) HMM1: sure they are the moment in the deluge

(49) HMM2: earthquake shivering uThI dharaa buildings vulnerable were they are the moment in the deluge

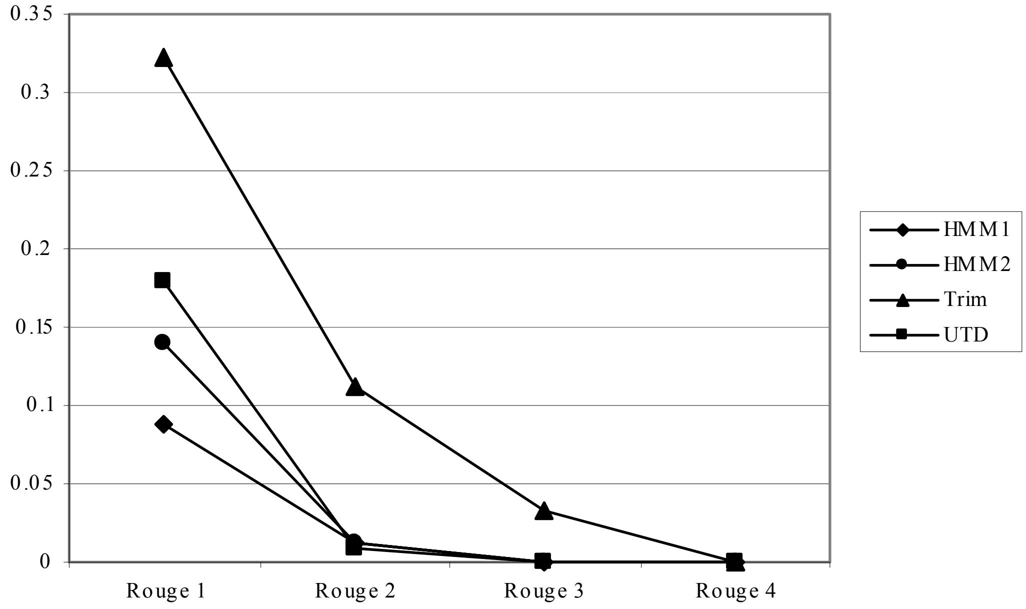


Figure 6.6: ROUGE Scores for UMD/BBN systems on Hindi Headlines

(50) Trimmer: gujarat earthquake tremble raised land unsafe buildings were earth

(51) UTD: people earthquake building india space columbia police kashmir team

Figure 6.6 shows the ROUGE scores of the four headline-generation systems. In this evaluation ROUGE was configured to use unigrams through 4-grams. The ROUGE scores show that Trimmer performs better than the other systems across n-gram size, and that there is no significant difference between the two HMM systems. However, at N=1, UTD scores better than the statistical systems. At N>1, there is no significant difference among the HMM and UTD systems.

The second evaluation involved human clarity judgments for the output of HMM2, Trimmer and UTD topic lists. Three subjects were shown the output of the three systems and asked them to rank the summaries from 1 (worst) to 5 (best) for whether they could tell what happened in the story based on the headline. The

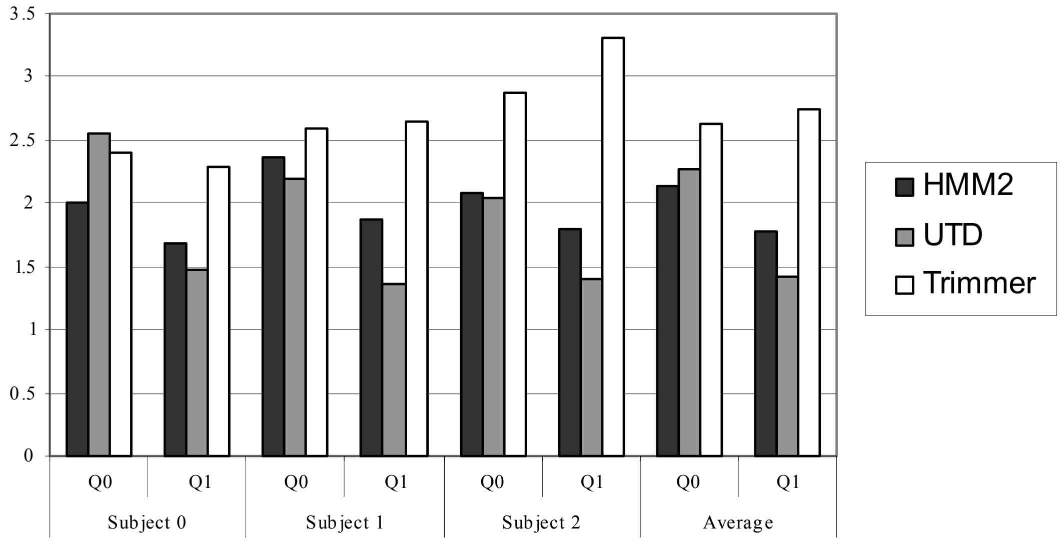


Figure 6.7: Human Evaluation of UMD systems for Hindi Headlines.
Q0: Can you tell what happens in the story based on the candidate summary?
Q1: Based on the reference summaries, was the candidate summary correct?

subjects were then shown the four reference summaries and asked to rank the correctness of the headlines on the same scale. The results of this evaluation are shown in Figure 6.7. The human judgments coincided with those of both automatic metrics: Trimmer performed better than the other systems and UTD was significantly lower in all three evaluations.

6.3.5 Spanish to English Summarization for CLIR

Trimmer was also used in University of Maryland's submission to the Interactive Track of the Cross-Language Evaluation Forum (iCLEF) 2003 (Dorr et al., 2003a). Experiments were performed to investigate a searcher's ability to recognize relevant documents based on English summaries of Spanish documents. Searchers were shown document surrogates and asked to decide if the document was relevant to a query based on the surrogate. The two types of surrogates tested were (1) a complete extract of the first 40 (translated) words of each news story (F40) and (2) Cross-Language Trimmer applied to a machine translation of the first sentence of the document (CLT). Eight subjects were shown surrogates for the results of IR searches on eight topics. The translations and search results were provided by iCLEF.

A sample query is shown in Sentence (52i). Example surrogates are shown in Sentences (52ii) and (52iii).

(52) (i) Find information about fishing quotas in the EU.
(ii) F40: The Portuguese shipbuilders of fishing affirmed, Tuesday today, that the fleet of high seas of its country, that works in the northeastern Atlantic, runs the risk of disappearing if they do not increase the fishing contingents
(iii) CLT: Portuguese shipbuilders of fishing affirmed that fleet of high seas runs risk

Each search result consisted of 50 documents. For each topic, the subjects were shown a description of the topic and surrogates for the 50 documents. The subjects were asked to judge whether the document was highly relevant, somewhat relevant or not relevant to the topic and whether they were highly confident, somewhat confident or not confident in their relevance judgment. The order of topics and whether the subject saw F40 or CLT for a particular topic was varied according to the Latin Square provided by iCLEF as part of the standard experiment design.

The goal of the experiment was to show that the two surrogates had close recall and precision, but that subjects shown CLT took less time to perform the task. The results show that subjects are able to perform the task faster using CLT,

	Cross-Language Trimmer	First 40
Total Judgment Count	1388	1189
Total Time (minutes)	272:37	290:34
Rate (judgments/minute)	4.09	5.09
Average Precision	0.4883	0.5939
Average Recall	0.2805	0.3769
Average $F_{\alpha=0.8}$	0.3798	0.4737
Average Pairwise κ	0.2704	0.2601

Table 6.6: Results of iCLEF 2003 cross-language summarization experiment. The two systems are Cross-Language Trimmer and a baseline consisting of the first 40 translated words.

but there with a substantial reduction in precision and recall. The results of the experiment are given in Table 6.6.

Inter-annotator agreement did not differ much between the two systems. Cohen's κ (Cohen, 1960) was used to measure the pairwise inter-annotator agreement. κ is 0 when the agreement between annotators is what would be expected by chance, and is 1 when there is perfect agreement. Due to the experiment design, it was not possible to calculate system-specific inter-annotator agreement for each pair of annotators because some pairs of annotators never used the same surrogate for judging the same documents. The average overall κ score for those cases in which subjects did see the same surrogate for the same document was 0.2455, while the average pairwise κ score for F40 was 0.2601 and the average pairwise κ score for CLT was 0.2704.

After the subjects completed judging the documents for a topic, they were asked the following questions:

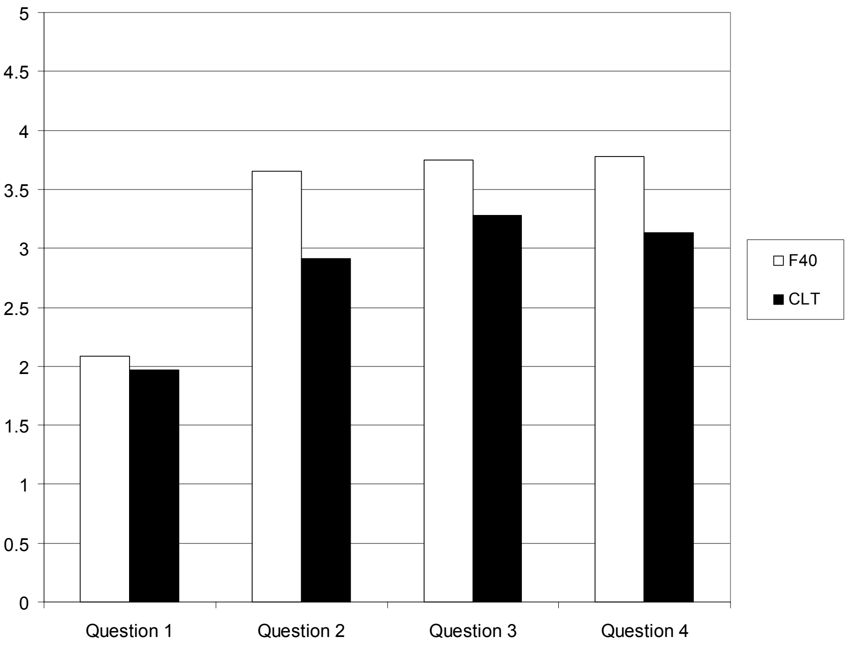


Figure 6.8: iCLEF2003 question responses for Spanish-English cross-language summarization

1. Were you familiar with this topic before the search?
2. Was it easy to guess what the document was about based on the surrogate?
3. Was it easy to make relevance judgments for this topic?
4. Do you have confidence in your judgments for this topic?

The subjects answered each question by selecting a number from 1 to 5, where 1 meant "not at all", 3 meant "somewhat" and 5 meant "extremely." The responses are shown in Table 6.8.

This result should not necessarily be taken to mean that informative headlines are worse surrogates than the first forty words. It is likely that the headlines used in CLT were not good enough headlines to make a conclusion about informative summaries in general. Also, the average length of the headlines used in CLT was

- LIST FACTS ABOUT EVENT [event]
- DESCRIBE THE PROSECUTION OF [person] FOR [crime]
- IDENTIFY PERSONS ARRESTED FROM [organization] AND GIVE THEIR NAME AND ROLE IN ORGANIZATION

Figure 6.9: Examples of GALE Distillation Query Templates

much shorter than 40 words, giving F40 the advantage of including more topic information.

6.4 Structured Queries

The University of Maryland collaborated with IBM and Carnegie Mellon University on the Rosetta Team submission to the GALE Year 1 Distillation Evaluation[2]. University of Maryland's contributed Trimmer and URA relevance and centrality scoring. The Distillation evaluation was part of the Global Autonomous Language Exploitation program sponsored by DARPA. The goal of the program is to promote development of systems that can provide integrated transcription, translation and distillation services to warfighters. In this context, distillation is a combination of information retrieval and summarization. The task is to produce a set of text segments, called snippets, in response to instances of query templates. Figure 6.9 shows some examples of query templates used in the evaluation. Compression 4 was actually chosen by the sentence selector for inclusion in the distillation.

The Rosetta Team Distillation architecture is shown in Figure 6.10. After

[2] http://projects.ldc.upenn.edu/gale/

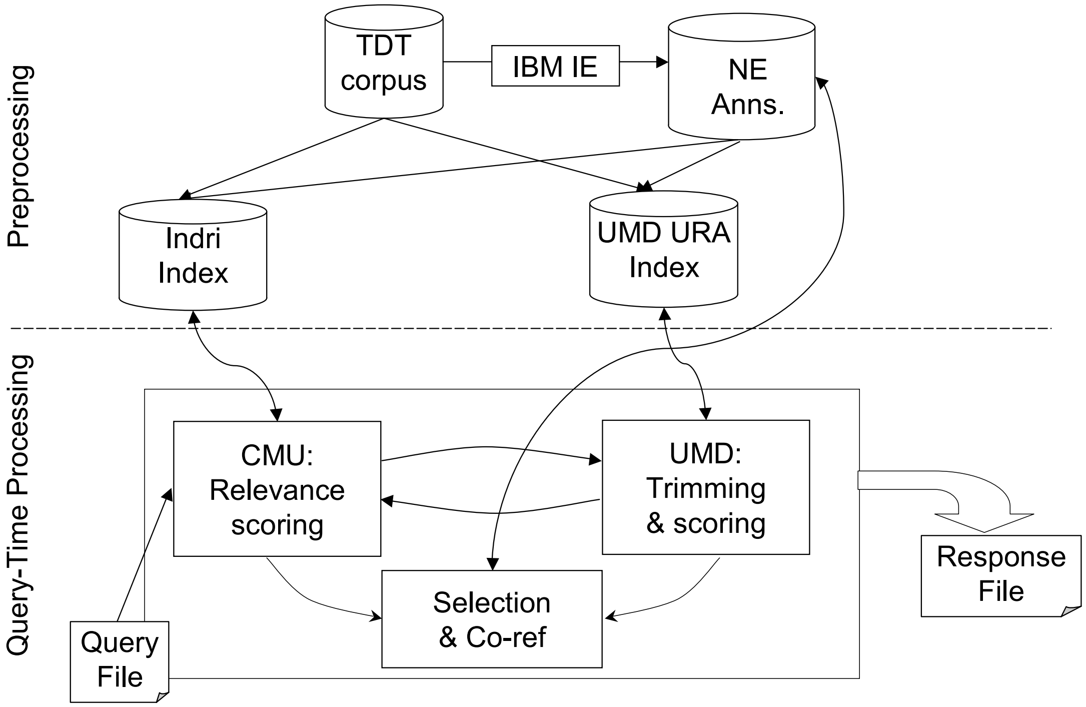


Figure 6.10: Rosetta Team Architecture for Distillation Task

snippets had been selected and scored by CMU, and co-reference resolution had been done by IBM, the snippets were sent to Trimmer which produced multiple candidates for each snippet. The candidates were then rescored by CMU and final sentence selection was performed by IBM. An example of a source snippet and the candidates created by Trimmer is shown in Figure 6.4.

Although there was no length constraint on the distillations, the evaluation process provided a motivation for sentence compression. Reviewers considered the density of relevant words in the snippets that provided new information. The re-

Source Snippet: In the future, China in the Asia-Pacific Economic Cooperation organization will continue to play a positive and constructive role.
Compression 1: In the future, China in the Asia-Pacific Economic Cooperation organization will continue to play a positive and constructive role.
Compression 2: In the future, China in the Asia will continue to play a positive and constructive role.
Compression 3: In the future, China will continue to play a positive and constructive role.
Compression 4: China in the Asia-Pacific Economic Cooperation organization will continue to play a positive and constructive role.
Compression 5: China in the Asia will continue to play a positive and constructive role.

Figure 6.11: An example of a source snippet and 5 compressions from a dry run of the Rosetta Distillation system. Compression 4 was selected by the sentence selector.

viewers evaluated the first 500 words of the system output. If they found the density higher than 0.5, they continued evaluating the system output in 200 word blocks until the density fell below 0.5. Thus compressed snippets in which non-query-relevant material has been removed would have increased the likelihood that the reviewer will continue reading the system output.

Due to system configuration problems, Trimmer was ultimately omitted from the Rosetta Team's distillation processing, and no internal evaluation was made of the effect of sentence compression on system performance. It remains as future research to determine the effect of sentence compression in the processing of structured queries.

6.5 Summary

This chapter described how MASC technology was applied to two new genres (broadcast news transcripts and email threads) and two new applications (cross-language summarization and structured queries). In the genre of broadcast news, Topiary was found to perform better than Trimmer, which mirrors the performance on written news. Email thread summarization was observed to be such a difficult problem that humans cannot do it well. In the area of cross-language summarization, Trimmer applied to machine translation was found to perform better than topic lists alone.

Chapter 7

Summary and Future Work

This chapter will present a summary of the dissertation, identify areas for future research and conclude.

7.1 Summary

This dissertation has explored the use of sentence compression as a tool for a variety of automatic summarization tasks under the MASC framework. The three stages of the MASC framework are sentence filtering, sentence compression, and sentence selection. Sentence compression in the context of extractive summarization addresses the limitations imposed by extracting at the granularity of the sentence. Sentences can contain a mixture of relevant and non-relevant information, or may be too large for the desired summary size. For summaries consisting of multiple sentences, overlapping content of otherwise important sentences becomes a concern. The MASC framework addresses these problems by generating multiple compressed versions of sentences from the source sentences, and constructing summaries from the candidates with a feature-based sentence selection tool.

Two sentence compression tools have been presented: HMM Hedge and Trimmer. In addition, Topiary generates candidates by combining topic terms with compressed text. These candidate generation tools have been applied to single-document and multi-document summarization tasks under the MASC framework in combination with a sentence selector that balances relevance against anti-redundancy. Across several domains and summarization tasks, sentence compression has been shown to improve system performance over systems that do not use any compression.

The following are among the important and interesting results reported in this dissertation:

- Humans can create fluent and informative summaries for news stories by selecting words in order from the story. (Section 3.1)
- A human extrinsic evaluation demonstrated that humans perform a relevance judgment task using summaries much faster than they could with a full text, but at a cost in precision and recall. (Section 4.7.1)
- A human study demonstrated that the use of sentence compression in multi-document summarization can potentially yield a 17% reduction in size without loss of relevance. (Section 5.1).
- Generation of multiple candidates with feature-based sentence selection improves Topiary beyond its state-of-the-art performance in DUC2004. (Section 4.6.1)
- An automatic evaluation using Rouge showed that systems using sentence compression for single-document summarization outperform comparable systems that do not use compression. (Section 4.6.3)
- An evaluation using Rouge showed the promise of using sentence compression under the MASC framework to improve system performance over a baseline that did not use sentence compression. (Section 5.4.4)
- An evaluation of single-document summarization on broadcast news transcripts found that use of sentence compression improved performance over a comparable system that did not use compression. (Section 6.1.4)
- An initial application of the MASC framework to email threads from the highly technical Enron corpus found that the task was so difficult that humans had a hard time interpreting and performing the summarization task. (Section 6.2.3)
- HMM Hedge performs better using blocks of words that respect sentence boundaries than arbitrary blocks of words, and best performance was achieved by selecting the first document sentence. (Section 3.4.2)

- HMM Hedge performs better by generating multiple compressions and selecting among them based on a linear combination of features than by generating the single most likely compression of a sentence. (Section 3.4.3)
- Topiary, a system that combines topic terms with compressed text, performs better than either topic terms or compressed text alone. (Section 4.6.3)
- Automatic evaluation tools give higher relative rankings to HMM Hedge than human evaluations because it produces compressions that are rich in important terms, but are difficult to read. (Section 4.7.1)
- In a multi-document summarization evaluation, a system that used Trimmer compression contained, on average, 2.73 more sentences per summary than a comparable system that did not use compression. (Section 5.5.6)

This dissertation yields the following contributions in the area of automatic summarization:

- Two sentence compression tools, HMM Hedge and Trimmer were implemented and evaluated on a variety of tasks. A third candidate generation tool, Topiary, was the top performing system in the DUC2004 short summary generation evaluation.
- Sentence compression was used as a tool to improve system performance on a variety of automatic summarization tasks.
- Multiple candidate sentence compression and feature-based candidate selection were combined to improve system performance on a variety of automatic summarization tasks.
- The MASC framework, consisting of sentence filtering, compression, and selection, was demonstrated as a framework for developing and evaluating automatic summarization tools.
- Headlinese, a dense yet easily comprehensible compressed version of English, was used as a model for the generation of short, informative summaries.
- Automatic summarization techniques developed under the MASC framework were applied to novel genres and applications, including broadcast news, email thread and cross-language summarization.

7.2 Future Directions

The MASC framework contains three stages: filtering, compression, and selection. Presently the only way to measure improvement is by examining the system output. Future improvements to MASC systems can be made by improving each stage separately. The filtering stage should be tested to ensure that important sentences are sent to the sentence compression tool. The compression tools should be evaluated to make sure that they generate the compressions that humans would want to select, and the sentence selector must make good use of space that compression makes available by choosing relevant and non-redundant candidates.

A summarization task that has not been explored in this work is multi-sentence summaries for single documents. This is a task that can be easily handled under the MASC framework.

Filtering, the selection of sentences in a document for subsequent processing, is an area of the MASC framework that has not yet been fully explored. Existing work in the area of sentence extraction without compression can be used as the filtering stage of a MASC system. This will be necessary to extend the work to summarization of informal genres, such as email, newsgroups, blogs, and transcribed casual speech which are not structured like news stories.

The decoding parameters for HMM Hedge should be optimized to produce the best compressions.These are difficult to tune because only a few of the generated compressions appear in a summary. One possible approach is to train the decoding parameters to maximize the sum of the rouge scores of all generated candidates.

Another aspect of HMM Hedge that can be improved is the morphological variation of verbs. The emit probabilities of the H states labeled with morphological variants of story verbs should be learned from the distribution of verb forms in headlines and stories.

Although the Trimmer rules have been designed to preserve important content, syntactic information alone is not enough to ensure this. The Trimmer rules could be expanded to make use of thematic role information to recognize parts of a sentence that should not be removed. Currently candidates are generated only by compressing a single sentence that occurs in the corpus. Anaphora resolution and co-referencing of entities could be used to improve the context content of sentences before compressing them. Sentence merging could also expand the pool of available candidates.

Multi-candidate rules make the Trimmer algorithm only partly order independent. Any Trimmer rule can be treated as a multi-candidate rule by either applying it to an applicable point in a tree or not. This may cause an explosion in the number of possible trimmed trees. More should be done to make the Trimmer algorithm fully order-independent.

However, there remains the question of why in some cases multi-candidate rules increased the number of candidates and yet performance did not improve. This is case in which the problem is the interaction of the candidate generator and the sentence selector, and deserves careful analysis.

Redundancy is currently detected only by word stem overlap. Paraphrase detection (Ibrahim et al., 2003; Shen et al., 2006) can be used to improve sentence selectors' ability to choose sentences that are non-redundant to each other.

The redundancy parameter λ should be optimized. One possibility is to indirectly optimize the performance of the summarizer by using the terms balanced by λ as separate dynamic features, and observing the relationship between the learned weights. Another is to develop a direct metric for redundancy and train λ to optimize that metric.

The parameters used by the sentence selector for multi-document summarization are currently learned by optimizing for Rouge on the system output. However, the value that should be optimized is the change in Rouge at each iteration of the sentence selector. In future development, the weights should be optimized to select among the candidates the one that will maximally increase the Rouge score of the current summary state. It may also be that these weights change as the iterations progress.

It is not unusual for a highly informative sentence to contain a pronoun referring to an entity from another sentence. Even if a Topiary topic covers the information gap, space is still wasted on the pronoun. Anaphora resolution of pronominal references will improve the pool of available sentences.[1]

MuD MASC provides a framework for progress in these areas. MuD MASC is capable of accepting multi-candidate compressions from different sources, as demonstrated by HMM-based and Trimmer-based MuD MASC. In addition, MuD MASC can make use of different combinations of features for making summary sentence selections, as shown by the use of Trimmer-internal features, HMM Hedge-internal fea-

[1] For example, using LingPipe http://www.alias-i.com/lingpipe/index.html

tures, URA relevance and centrality scores and IDA/CCS's estimated oracle score.

None of the systems discussed take sentence ordering into consideration. Sentence ordering can have an effect on structure and coherence of multi-document summarizations. Recent work in this area includes Conroy et al. (2006b); Barzilay et al. (2002); Okazaki et al. (2004) and Lapata (2003).

Evaluation of summarization systems remains a difficult problem. HMM Hedge was generally rated higher by automatic evaluation tools than by human performance of an extrinsic task. The MASC framework is a good tool for testing the correlation of automatic evaluation of summarization and human performance under different conditions.

7.3 Conclusion

The primary motivation for this research has been the observation that summarization by extraction at the granularity of the sentence is a limitation in making best use of limited space. This research has shown that summarization systems using sentence compression can outperform comparable systems that are limited to source sentences. The combination of multiple alternative sentence compressions with feature-based candidate selection is the framework under which the usefulness of sentence compression becomes apparent.

Appendix A

Implementation of HMM Hedge Sentence Compression

HMM Hedge takes as input a string of tokens T representing a block of words from a document. The blocks of words typically correspond to sentences, but this is not required. The tokens representing verbs are tagged as verbs. The first and last tokens are special start and end tokens. Punctuation tokens are not included in T. The output is a set of subsets of tokens from T that represent compressions of T. Each compression is associated with a score. The tokens in the subsets are constrained to appear in the same relative order as the tokens in T. Each subset is required to contain at least one verb. The subsets may contain morphological variants of verbs in T.

The unigram and bigram probabilities of tokens are calculated using separate language models of headlines and newspaper stories, referred to henceforth as the headline model and the story model. For each word w in a corpus the following values are stored:

- count: The number of times w appears in the corpus.
- dfcount: The number of distinct following words observed following w in the corpus.
- A measure of the diversity of w's set of following words: $\lambda = \frac{dfcount}{dfcount+count}$.

For each pair of words w_n, w_{n+1} in a corpus, the number of times the bigram appears in the corpus is stored. The first word of a sentence is counted as following the start token and the end token is counted as following the last word of a sentence.

The probability calculations also use C_{token}, the number of tokens in the corpus, $C_{distinct}$, the number of distinct words in the corpus, and $C_\lambda = \frac{C_{distinct}}{C_{distinct}+C_{token}}$.

The unigram probability of w is calculated by

$$unigram(w) = \begin{cases} (1 - C_\lambda)\frac{w_{count}}{C_{token}} + C_\lambda \frac{1}{C_{distinct}} & \text{if w appears in C} \\ C_\lambda \frac{1}{C_{distinct}} & \text{otherwise} \end{cases}$$

The bigram probability of bigram v, w is calculated by

$$bigram(v, w) = \begin{cases} (1 - v_\lambda)\frac{w_{count}}{C_{token}} + C_\lambda \frac{1}{C_{distinct}} & \text{if v and w appear in C} \\ v_\lambda C_\lambda \frac{1}{C_{distinct}} & \text{if v appears in C and w does not} \\ (1 - C_\lambda)\frac{w_{count}}{C_{token}} + C_\lambda \frac{1}{C_{distinct}} & \text{if w appears in C and v does not} \\ C_\lambda \frac{1}{C_{distinct}} & \text{if neither v nor w appear in C} \end{cases}$$

A dynamic programming algorithm is used to find the highest scoring compressions of T over a range of compression lengths. The scores are based on the following compression features:

- The bigram probabilities in a headline model of the words in the compression. In the noisy channel model, this is the probability that the compression occurs as a headline.

- The unigram probabilities in a story model of the words in T that are not in the compression. This is the probability that the noisy channel alters the compression to form T by adding the non-compression words.

- The probabilities that the words in the compression emit the corresponding words in T. This is the probability that the noisy channel alters a word from the compression to the form that appears in T, for example *announces* to *announced*.

- The sum of the positions in T of the words in the compression. For example if a compression consists of T[1], T[4], and T[8] the sum of the positions is $1 + 4 + 8 = 13$.

- The number of one-word gaps in the compression. For example, if the compression contains T[6] followed by T[8] it is a one-word gap.
- The number of multi-word gaps in the compression. For example, if the compression contains T[6] followed by T[12] it is a multi-word gap.
- The number of strings of contiguous words, or clumps, in the compression. For example, T[6],T[8], and T[9] are a contiguous clump of words. A compression consisting of T[6],T[7],T[8],T[11],T[12],T[13] contains two clumps.

The following parameters affect the behavior of HMM Hedge.

- MAXLEN: The length in tokens of the longest compressions to be produced. The default value is 15.
- MINLEN: The length in tokens of the shortest compressions to be produced. The default value is 5.
- N_REPORT: The number of compressions to produce at each length from MINLEN to MAXLEN. The default value is 5.
- N_BEST: The number of back pointers to keep at each cell of the dynamic programming array. Keeping multiple back pointers allows the decoder to find the n-best solutions rather than a single best solution. A larger number of back pointers comes closer to finding the optimal n-best solutions, however it is expensive in time and memory. The default value is 3.
- POSITION_BIAS_FACTOR: Let POSITION_BIAS_FACTOR = B. For each token T[n] in the compression the score is multiplied by B^n. When $B < 1$, it provides a bias against compressions using tokens farther from the start of T. The default value is 0.95.
- SINGLE_WORD_GAP_BIAS: Let SINGLE_WORD_GAP_BIAS = B. For a candidate with n single word gaps, the score is multiplied by B^n. When $B < 1$, it provides a bias against compressions with many single word gaps. The default value is 0.99.
- MULTI_WORD_GAP_BIAS: Let MULTI_WORD_GAP_BIAS = B. For a candidate with n multi-word gaps, the score is multiplied by B^n. When $B < 1$, it provides a bias against compressions with many multi-word gaps. The default value is 0.95.
- CLUMP_BIAS: Let CLUMP_BIAS = B. For a candidate with n clumps of contiguous tokens, the score is multiplied by B^n. When $B < 1$, it provides a bias against compressions with many small clumps in favor of compressions with fewer longer clumps. The default value is 0.95.

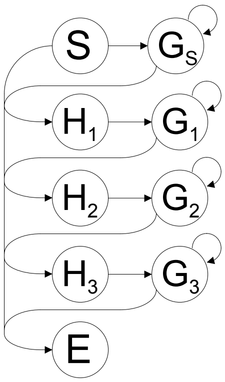


Figure A.1: HMM for a Story Containing Three Words. S represents the start state and E the end state. The H states emit story words that appear in the headline and G states emit story words that do not appear in the headline. The G state associated with an H state emits all the words that occur in the story between that headline word and the next word in the headline.

The compression of T is represented by a noisy channel model in which an unseen compression is passed through a noisy channel with T as the observed output. The transmission is represented by an HMM similar to the one shown in Figure A.1. Each path through the HMM represents a compression of T. The words in T are emitted by the HMM, with the words that appear in the compression emitted by H states and all other words emitted by G states.

Morphological variation of verbs is handled by creating multiple H states for each verb in T. For example, the verb *blow* in T could be emitted by H states labeled *blow, blows, blowing, blew,* or *blown.* For non-verbs the probability that an H state emits it's corresponding word in T is 1.0. For verbs, the probability is $\frac{1}{N}$ where N is the number of H states corresponding to the word in T, that is, the number of distinct verb forms.

Each path through the HMM is associated with a score. For ease of calculation, a path's score is calculated using sums of logs. Consider a path P through the states of an HMM containing a mixture of H states and G state. The cost of the path is the sum of the following values:

- The sum over the H states in P of the log of the bigram probability in the headline model of v, w, where w is the word associated with the H state, and v is the word associated with the previous H state in the path.
- The sum over the H states of the position bias, $n\ log(B_{position})$, where n is the position in T of the H state's corresponding token in T.
- The sum over the G states in P of the log of the unigram probability in the story model of w, the token in T emitted by the G state.
- The sum of the gap biases. Every time the path spends time in a G state, the a gap bias is added to the score. If the path emits only one token in T before leaving the G state, the log of the single-word gap bias is added. If more than one token in T is emitted before leaving the G state, the log of the multi-word gap bias is added.
- The sum of the clump biases. The number of clumps in a path is equal to the number of transitions from an H state other than the start state to a G state or the End state. In other words, clumps are counted when the end, so the transition from the start state to a G state doesn't count, and the transition from an H state to the end state does.

The dynamic programming problem is for each length N from MINLEN to MAXLEN, find the path from the start state to the end state that emits the tokens in T and passes through exactly N H states with the highest score. The problem of finding the highest score path to a state with N H states can be decomposed into smaller problems. For a G state g, the problem is to find the state s that maximizes the score of the highest scoring path to s with N H states plus the cost of transitioning from s to g. For an H state h, the problem is to find the state s that maximizes the score of the highest scoring path to s with $N - 1$ H states plus the cost of transitioning from s to h.

The workspace for the dynamic programming problem is a three-dimensional array. The three axes are T, S, and L. T represents the tokens in the source sentence, S represents the states of the HMM and L represents the length in tokens of the compression. Thus the cell at coordinates $[t, s, l]$ contains the scores and backpointers for the N_BEST best paths from the start state to state s, such that the paths contain l H states and emit tokens T[1] through T[t-1]. The paths are constrained to emit the tokens in T, so the T axis functions as a clock: the scores of the cells with $T = t$ can be calculated from the values in earlier cells, i.e., those with $T < t$.

The function for calculating the score of cell $[t_2, s_2, l_2]$ when transitioning from cell $[t_1, s_1, l_1]$ is as follows. Let $cost(t, s, l)$ be the cost of the best path containing l H states to state s emitting token t, $bigram\_prob(s_1, s_2)$ be the bigram probability in the headline model of the words associated with states s_1 and s_2, $emit\_prob(s, t)$ be the probability that state s emits T[t], and $unigram\_prob(t)$ be the unigram probability in the story model of T[t]. Table A.1 shows the formulas for calculating the score of a cell depending on the type of transition. Back pointers are stored in each cell for the N_BEST best paths to the start state.

The final step of HMM Hedge compression is decoding or traceback, in which the highest scoring paths are found and expressed as subsets of tokens from T. The highest scoring path is found by starting from the cell representing the end state and a path containing the desired number of H states, and stepping backward through the highest-scoring back pointer in each cell. The words associated with the H states in this path constitute the compression.

The N_REPORT best paths are found by maintaining a list of the N_REPORT best paths during the traceback process. At each cell the tracebacks other than the best traceback are considered. During traceback the score of the best start-to-end

From	To	Score
Start	G State	$0 + log(unigram\_prob(t_2))$
Start	H State	$0 + log(bigram\_prob(s_1, s_2)) + log(emit\_prob(s_2, t_2)) + t_2\ log(B_{position})$
H State	H State	$cost(t_1, s_1, l_1) + log(bigram\_prob(s_1, s_2)) + log(emit\_prob(s_2, t_2)) + t_2\ log(B_{position})$
H State	G State	$cost(t_1, s_1, l_1) + log(unigram\_prob(t_2)) + log(B_{clump})$
G State	H State	$cost(t_1, s_1, l_1) + log(bigram\_prob(s_1, s_2)) + log(B_{gap}) + log(emit\_prob(s_2, t_2)) + t_2\ log(B_{position})$
G State	G State	$cost(t_1, s_1, l_1) + log(unigram\_prob(t_2))$
G State	End	$cost(t_1, s_1, l_1) + log(bigram\_prob(s_1, s_2)) + log(B_{gap})$
H State	End	$cost(t_1, s_1, l_1) + log(bigram\_prob(s_1, s_2)) + log(B_{clump})$

Table A.1: Formulas for calculating cell scores.

path passing through a state s is known. It is also possible to calculate the difference between the scores of the best start-to-s path and next-best start-to-s path. The score of the start-to-end path using the next-best start-to-s path is the score of the best start-to-end path minus the difference between the scores of the best start-to-s path and next-best start-to-s path. If this score is better than any of the paths in the best paths list, it is added to the best paths list and the lowest-scoring best path is dropped.

Paths that do not contain a verb are automatically given scores of $-\infty$. If a complete traceback is performed and no compressions containing a verb were found, this constraint is removed and the N_REPORT best compressions are reported even though they do not contain a verb.

Appendix B

Implementation of Trimmer Sentence Compression

Trimmer takes as input a sentence from a document and a desired maximum length for the compression. The sentence is pre-processed by creating a parse tree that is compatible with the Penn Treebank conventions and by tagging the words in the sentence that are parts of named entities or time expressions. The output is a either a set of compressions or a single compression of the sentence. Compression is done by applying syntactic trimming rules, Trimmer rules, to the parse tree.

Define the surface string of a parse tree to be a string containing the tokens from the leaves of the parse tree with appropriate whitespace. There are two basic operations that remove syntactic components: mask and mask outside. When a Trimmer rule *masks* a node n in a parse tree the descendants of n do not appear in the surface string of the parse tree. When a Trimmer rule *masks outside* a node n, the nodes that are not descendants of n do not appear in the surface string of the parse tree. Single candidate Trimmer rules consist of one or more operations, applied together as a single action. A single compression for a sentence is generated by applying Trimmer rules until the surface string is shorter than the maximum length. Multiple compressions are generated by setting the maximum length very low, for example 15 characters, and taking the surface string of the parse tree after each application of each Trimmer rule action.

Multi-candidate rules generate multiple compressions, each of which can serve a starting point for the application of other rules. These rules increase the pool of compressions by producing each option when there is a choice of how to apply a rule. For example, the single candidate rule for conjunctions can mask the right or left child, but only one action is taken, producing a single output. The multi-candidate rule produces three outputs in which the right child, the left child and neither child are masked.

Trimmer uses Charniak's parser (Charniak, 2000) for pre-processing. This appendix will describe rule triggers and actions with respect to tags and features of Charniak's parser. The named entity tagging is done by IdentiFinder (Bikel et al., 1999), and the named entity and time expressions mentioned in this appendix are those assigned by IdentiFinder.

B.1 Single Candidate Trimmer Rules

Each single candidate Trimmer rule consists of a pattern to search for in a parse tree, and an action to take, consisting of mask and/or mask outside operations. A trimmer rule is applied by searching through the tree to find all the nodes that match the rule's pattern. The rule's action is applied one-by-one to the nodes in the parse tree that match the pattern. The actions are taken in the reverse order of a depth-first traversal of the parse tree.

The Trimmer rules are applied in the following order:

1. Select Root S node
2. Remove temporal expressions

3. Remove some determiners
4. Remove possessive pronouns
5. Remove modal verbs
6. Remove auxiliary verbs
7. Remove complementizer *that*
8. Apply the XP over XP rule
9. Remove conjunctions
10. Remove preposed adjuncts
11. Remove PPs that do not contain named entities
12. Remove SBARSs
13. Remove all PPs under SBARs
14. Backtrack to state before removing PPs
15. Remove SBARs
16. Remove PPs that do not contain named entities
17. Remove all PPs

The remainder of this section will describe the patterns and actions of the single candidate Trimmer rules.

The pattern for the root S selection rule matches any node that has part of speech (PoS) tag S, has a child with PoS tag NP, and has a subsequent child with PoS tag VP. The Root S rule chooses the deepest matching node in the tree such that it and all of its ancestors are leftmost children of a node with PoS tag S (or the TOP node). The action for the rule is a single mask outside operation on the chosen node.

There are 2 patterns for the temporal expression rule. For a node j, let n be j's parent, and o be n's parent. If j is a leaf node containing text that has been marked by IdentiFinder with a TIMEX tag, n has PoS NP and o has PoS PP, then a mask operation is applied to node o. This removes *on Sunday* from sentences like "John said on Sunday that he would attend." If j is a leaf node containing text that has been marked by IdentiFinder with a TIMEX tag and n has PoS NP, then a mask operation is applied to node n. This removes *Sunday* from sentences like "John said Sunday that he would attend."

The pattern for the determiner rule matches a node if it has PoS DT and the surface word associated with the node is *the*, *a*, or *an*. A mask operation is applied to the node.

The pattern for the possessive pronoun rule matches a node if it has PoS PRP$. A mask operation is applied to the node.

The pattern for the modal be rule matches a node n if it has PoS VP, has a child m with PoS MD, has another child o with PoS VP, and o has a child p with PoS VB and surface word *be* or *have*. The action is to mask nodes m and p. This removes, for example, *will be* from "John will be making the bed." A secondary pattern in the rule matches if the parent of n has PoS S, and the parent has a child q with PoS NP, and q has a child r with surface word *there* or *it*. The secondary action masks node r. This removes, for example, *It will be* from "It will be raining in Baltimore."

The pattern for the auxiliary verb rule matches a node n with PoS VP, a child o with PoS AUX, and a subsequent child p with PoS VP. The action is to mask o and any children of n between o and p, but not p. However, if any node between o and p has surface word *not*, it is not masked, and if any node between o and p has surface word *n't*, the entire rule is abandoned. This removes, for example, *is still* from "John is still looking for his dog," but will not affect "John isn't looking for a cat."

The pattern for the complementizer matches a node n with surface word *that* and PoS IN. The action is to mask n. This removes, for example, *that* from "Manuelo said that Luiz was a hero," but would not affect "Shulamit knows that poem by heart."

The pattern for the XP over XP rule matches a node n when n has PoS NP or VP, o is the first unmasked child of n, o has the same PoS as n, and o is the head child of n, as described in Section 4.2.1. The operation is to mask all children of n that come after o, but not o. However, if a child has PoS PP it is not masked, and if a child has PoS CC, it and subsequent children are not masked. This removes, for example, *who wore a hat* from "The man who wore a hat smoked a cigar," and *an action that earned a detention* from "Larry spat, an action that earned a detention."

The pattern for the conjunction rule matches a node n when n has PoS CC, m is the left sibling of n, o is the right sibling of n, and m and o have PoS VP. If n has surface word *and* or *or*, the action is to mask n and o. If n has surface word *but*, the action is to mask n and m. This removes, for example, *and ate a cookie'* from "Susan sang a song and ate a cookie," and *lost the bet but* from "Nancy lost the bet but learned a lesson."

The pattern for the preamble rule matches a node n when n is the root S node found by the root S rule, n has a child o with PoS NP, and o is the leftmost child of n with PoS NP. The action of the rule is to mask any children of n that are to the left of *o*. This removes, for example, *According to Chicken Little* from "According to Chicken Little, the sky is falling."

The pattern for the prepositional phrase rule matches a node n when n has PoS PP. The action is to mask n. The prepositional phrase rule can be applied with or without named entity protection. If named entity protection is enabled, the action will not happen if any of the descendant leaf nodes have surface words that have been tagged by IdentiFinder as ENAMEX. The types of ENAMEX include animal, contact info, disease, event, game, language, location, nationality, organization, person, plant, product, and substance.

In the Trimmer algorithm, the prepositional phrase rule is first applied with named entity protection enabled. It is run a second time with named entity protection disabled for prepositional phrases that are descendants of any node with PoS SBAR, because these constituents will be removed by the SBAR rule in the next stage anyway.

The pattern for the SBAR rule matches a node n when n has PoS SBAR. The action is to mask n. The SBAR removes subordinate clauses.

If, after the SBAR rule has been applied, the length of the surface string is still above the maximum length, the state of the tree before the prepositional phrase rule is restored. The prepositional phrase rule is applied to the entire tree with named entity protection disabled, followed by a repetition of the SBAR rule.

If, after all rules have been applied, the length of the surface string is still above the maximum length, then the remaining surface string is truncated to the maximum length.

B.2 Multi-Candidate Trimmer Rules

Three rules have been modified to produce multiple different output trees rather than a single output tree. Each output from a multi-candidate rule is the input to other Trimmer rules. After the three multi-candidate rule have produced a set of starting point trees, the single candidate rules are applied to the starting point trees in a fixed order, shown below. The candidates are the surface strings of the intermediate stages of the parse tree after the application of each multi-candidate and single candidate rule.

1. TIMEX Rule
2. Determiner Rule
3. Possessive Pronoun Rule
4. Modal Be/Have Rule
5. Auxiliary Verb Rule
6. Complementizer Rule
7. XP Over XP Rule
8. PP Rule with Named Entity Protection
9. PP Rule without Named Entity Protection
10. SBAR Rule

The remainder of this section will describe the patterns and actions of the multi-candidate Trimmer rules.

The pattern for the multi-candidate root S selection rule matches any node that has part of speech (PoS) tag S, has a child with PoS tag NP, and has a subsequent child with PoS tag VP. The action of the multi-candidate root S rule is to produce an output tree for each possible root S node with a mask outside operation applied to the root S node.

The pattern for the multi-candidate preamble rule matches all nodes n such that n is a possible root S node found by the root S rule, n has a child o with PoS NP, and o is the leftmost child of n with PoS NP. Note that it doesn't matter if n actually is the root S node of the tree or not, as long as it is a possible root S node. The action of the rule is to mask children of n to the left of o. An output tree is produced for each combination of possible applications of the preamble action, including a tree with no applications of the preamble action. Post-processing is used to ensure that trees with identical surface strings are not produced as output.

The pattern for the multi-candidate conjunction rule matches a node n when n has PoS CC, m is the left sibling of n, o is the right sibling of n, and m and o have PoS VP. For each matching node, there are three possible actions: mask n and m, mask n and o, or mask neither pair. Output trees are generated with all combinations of these three actions over all matching nodes in the tree. Post-processing is used to ensure that trees with identical surface strings are not produced as output.

B.3 Trimmer Features

Each Trimmer compression of a sentence is associated with a set of Trimmer-specific features that can be used in the candidate selection stage of MASC. These include the depth in the tree of the root S node, and the number of preamble, conjunction, time expression, determiner, possessive pronoun, auxiliary verb, modal be, complementizer, xp over xp, prepositional phrase with named entity protection enabled, prepositional phrase with named entity disabled, and SBAR actions were applied.

Appendix C

Implementation of Topiary Candidate Generation

Topiary takes as input a set of compressions of a sentence generated by Trimmer, a set of topic terms with scores generated by UTD and assigned to the document from which the sentence came by OnTopic, a maximum length for the candidates, and a requested number of topic terms. The output is either a set of candidates or a single candidate combining the topic terms with the Trimmer sentence compressions.

A single candidate is produced by looping through the Trimmer compressions from the longest to the shortest. Let T be the requested number of topic terms, and L be the maximum length. For each Trimmer compression, find the T highest scoring topic terms that do not occur in the compression. Combine the topic terms and compression by concatenating them together (topic terms followed by Trimmer compression) with a single space (or colon for easier human readability) between them. If the combination is below the maximum length, exit the loop. Otherwise continue to the next compression.

If there is space remaining under the maximum length for additional topic terms, add the highest scoring topic term that does not already appear in the candidate and for which there is room below the maximum length. Additional topic terms are added after the initially selected topic terms and before the separating white space (or colon).

If the loop reaches the final Trimmer compression, and no candidate has been found that is shorter than the maximum length, construct a candidate by concatenating the T highest scoring topic terms that do not occur in the shortest compression and the shortest compression, and truncate it to the maximum length.

Multiple Topiary candidates are generated by concatenating all combinations of non-redundant topic terms with each Trimmer compression. For each Trimmer compression, create a list of topic terms that do not occur in the compression. For each subset of non-redundant topic terms (including the empty set) combine the topic terms with the compression and truncate to the maximum length.

The multiple candidates from Topiary are associated with all the Trimmer features described in Appendix B and additional topic-related features. These include the number of topic terms, the sum of the scores of the topic terms, and the space taken up by the topic terms. These features can be used in the candidate selection stage of the MASC framework.

Appendix D

Implementation of MASC Sentence Selector

The MASC sentence selector takes as input a set of candidates with associated feature values, a set of learned feature weights, and a maximum length. The output is either a single candidate or a set of candidates that form a summary.

The single-sentence selector is straightforward. Each candidate is given a score which is the linear combination of the feature values and feature weights. The features include directly observable characteristics of the candidates (such as length), features derived from the compression processing, and IR scores calculated separately by URA. The candidate with the highest score is selected as the summary. The weights for the features are learned using BBN's Optimizer. ROUGE is used to score each candidate on a set of test data. Optimizer takes as input the candidate features and ROUGE scores and calculates a set of weights that maximizes the overall ROUGE score on the training corpus when used to select candidates.

Multi-sentence summaries are constructed by iteratively selecting the highest-scoring candidate based on a set of static and dynamic features. The static feature values are assigned before iteration begins, and are fixed for the duration of iteration. These include directly observable features, features derived from compression processing and IR scores. The dynamic scores change values at each iteration of the sentence selector. These scores change as a result of the changing status of the summary under construction. Their purpose is to balance relevance with anti-redundancy, and they include a word-overlap redundancy metric described in Section 5.3.2, and a count of how many candidates in the current summary are from the same document as each remaining candidate in the pool. After each iteration, the dynamic feature values of the available candidates are recalculated, and each candidate is given a score that is a linear combination of the static and dynamic features. A candidate is added to the summary with each iteration until the length of the summary exceeds the maximum length. After a candidate is added to the summary, all other compressions or variants derived the same source sentence are removed from the candidate pool. The candidate added at the final iteration is truncated so that the entire summary does not exceed the maximum length.

For the evaluations presented in this dissertation, the feature weights were learned by manually optimizing them to maximize the average Rouge scores of the final summaries on a training corpus. Recent work has shown that using BBN's Optimizer to learn weights that maximize δ-Rouge at each iteration gives a significant improvement in Rouge scores. δ-Rouge is a way of using automatic evaluation to score candidates rather than complete summaries. It is the change in Rouge score of a multi-sentence summary caused by adding a specific candidate.

After the summary candidates are selected, they are presented in the order they were added to the summary. As a final presentation stage, candidates may be annotated with meta-data, such as time-stamps or author names.

Bibliography

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190, 1983.

Michele Banko, Vibhu Mittal, and Michael Witbrock. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL 2000)*, pages 318–325, Hong Kong, 2000.

Regina Barzilay, Noemie Elhadad, and Kathleen McKeown. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55, 2002.

Leonard E. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3:1–8, 1972.

Sabine Bergler, René Witte, Michelle Khalife, Zhuoyan Li, and Frank Rudzicz. Using knowledge-poor coreference resolution for text summarization. In *Proceedings of the HLT-NAACL 2003 Text Summarization Workshop and Document Understanding Conference (DUC 2003)*, pages 85–92, Edmonton, Alberta, 2003.

Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1/3):211–231, 1999.

Sasha Blair-Goldensohn, David Evans, Vasileios Hatzivassiloglou, Kathleen McKeown, Ani Nenkova, Rebecca Passonneau, Barry Schiffman, Andrew Schlaikjer, Advaith Siddharthan, and Sergey Siegelman. Columbia University at DUC 2004. In *Proceedings of the 2004 Document Understanding Conference (DUC 2004) at HLT/NAACL 2004*, pages 23–30, Boston, Massachusetts, 2004.

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2): 79–85, 1990.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 335–336, Melbourne, Australia, 1998.

Eugene M. Charniak. A Maximum-Entropy-Inspired Parser. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2000)*, pages 132–139, Seattle, Washington, 2000.

Noam A. Chomsky. *Lectures on Government and Binding.* Foris Publications, Dordrecht, Holland, 1981.

Heidi Christensen, BalaKrishna Kolluru, Yoshihiko Gotoh, and Steve Renals. From text summarisation to style-specific summarisation for broadcast news. In *Proceedings of the 26th European Conference on Information Retrieval (ECIR04)*, Sunderland, U.K, 2004.

Heidi Christensen, BalaKrishna Kolluru, Yoshihiko Gotoh, and Steve Renals. Maximum entropy segmentation of broadcast news. In *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, Pennsylvania, 2005.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measures*, 20:37–46, 1960.

Thomas Colthurst, Owen Kimball, Fred Richardson, Han Shu, Chuck Wooters, Rukmini Iyer, and Herbert Gish. The 2000 BBN Byblos LVCSR system, 2000. URL `citeseer.ist.psu.edu/423053.html`.

John Conroy and Dianne P. O'Leary. Text summarization via hidden markov models and pivoted qr matrix decomposition. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans, Louisiana, 2001.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING/ACL 2006)*, Sydney, Australia, 2006a.

John M. Conroy, Judith D. Schlesinger, Dianne P. O'Leary, and J. Goldstein. Back to basics: CLASSY 2006. In *Proceedings of the 2006 Document Understanding Conference (DUC 2006) at HLT/NAACL 2006*, New York, New York, 2006b.

John M. Conroy, Judith D. Schlesinger, and Jade Goldstein Stewart. CLASSY query-based multi-document summarization. In *Proceedings of the 2005 Document Understanding Conference (DUC-2005) at NLT/EMNLP 2005*, Vancouver, Canada, 2005.

Nick Craswell, Arjen P. de Vries, and Ian Soboroff. Overview of the TREC-2005 enterprise track. In *Proceedings of TREC 2005*, 2005.

Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy, 1992.

Hoa Dang and Donna Harman. *Proceedings of the 2006 Document Understanding Conference (DUC 2006) at HLT/NAACL 2006.* 2006.

Hal Daumé and Daniel Marcu. Bayesian multi-document summarization at MSE. In *Proceedings of the MSE2005 Track of the ACL 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, Ann Arbor, Michigan, 2005.

Christopher P. Diehl, Lise Getoor, and Galileo Namataz. Name reference resolution in organizational email archives. In *Proceedings of the 2006 SIAM Conference on Data Mining*, 2006.

Jana Diesner, Terrill Frantz, and Kathleen Carley. Communication networks from the Enron email corpus. It's always about the people. Enron is no different. *Journal of Computational and Mathematical Organization Theory*, 11(3):201–228, 2005.

Bonnie J. Dorr, Daquing He, Jun Luo, Douglas W. Oard, Richard Schwartz, Jianqiang Wang, and David M. Zajic. iCLEF 2003 at Maryland: Translation selection and document selection. In *Proceedings of the Cross Language Evaluation Forum Workshop (CLEF 2003)*, Trondheim, Norway, 2003a.

Bonnie J. Dorr, Christof Monz, Stacy President, Richard Schwartz, and David M. Zajic. A Methodology of Extrinsic Evaluation of Text Summarization: Does ROUGE Correlate? In *Proceedings of the Association for Computational Linguistics (ACL) Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, Ann Arbor, MI, 2005.

Bonnie J. Dorr, David M. Zajic, and Richard Schwartz. Cross-language headline generation for Hindi. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(3):270–289, 2003b.

Susan Dumais, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. Stuff I've Seen: A system for personal information retrieval and re-use. In *Proceedings of the 26th Annual Conference of the Special Interest Group on Information Retrieval (SIGIR 2003)*, Toronto, Canada, 2003.

Ted Dunning. Statistical identification of language. Technical Report MCCS 94-273, New Mexico State University, 1994.

Timm Euler. Tailoring text using topic words: Selection and compression. In *Proceedings of 13th International Workshop on Database and Expert Systems Applications (DEXA 2002)*, pages 215–222, Aix-en-Provence, France, 2002.

Jade Goldstein, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of the ANLP/NAACL 2000 Workshop on Automatic Summarization*, pages 40–48, 2000.

Gregory Grefenstette. Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In *Intelligent Text Summarization, AAAI Spring Symposium Series, Stanford, California*, pages 111–117, 1998.

Donna Harman. *Proceedings of the 2004 Document Understanding Conference (DUC 2004)*. Boston, Massachusetts, 2004.

Donna Harman and Mark Liberman. TIPSTER complete. Linguistic Data Consortium (LDC), Philadelphia, 1993.

Daqing He, Jianqiang Wang, Jun Luo, and Douglas W. Oard. iCLEF 2004 at Maryland: Summarization design for interactive cross-language question answering. In *Proceedings of the Cross Language Evaluation Forum Workshop (CLEF 2004)*, Bath, United Kingdom, 2004.

Stacy President Hobson, Bonnie J. Dorr, and Christof Monz. Task-based evaluation of text summarization using relevance prediction. *Information Processing and Management Special Issue on Summarization*, 43, 2007.

Chiori Hori, Sadaoki Furui, Rob Malkin, Hua Yu, and Alex Waibel. Automatic speech summarization applied to English broadcast news speech. In *Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP) 2002*, Orlando, Florida, 2002.

Ali Ibrahim, Boris Katz, and Jimmy Lin. Extracting structural paraphrases from aligned monolingual corpora. In *Proceedings of the Second International Workshop on Paraphrasing (ACL2003)*, pages 50–57, Sapporo, Japan, 2003.

Rong Jin and Alexander G. Hauptmann. Automatic title generation for spoken broadcast news. In *Proceedings of the First International Conference on Human Language Technology Research*, San Diego, California, 2001.

Hongyan Jing. Sentence reduction for automatic text summarization. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP'00, Seattle, Washington)*, 2000.

Hongyan Jing, Regina Barzilay, Kathleen McKeown, and Michael Elhadad. Summarization evaluation methods: Experiments and analysis. In *Proceedings of the AAAI Symposium on Intelligent Summarization*, Stanford University, CA, March 23-25 1998.

Hongyan Jing and Kathleen McKeown. The decomposition of human-written summary sentences. In *Proceedings of teh 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 129–136, Berkely, California, 1999.

Hongyan Jing and Kathleen McKeown. Cut and paste based text summarization. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2000)*, pages 178–185, Seattle, Washington, 2000.

Kevin Knight and Daniel Marcu. Statistics-based summarization—step one: Sentence compression. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, Austin, Texas, 2000.

Kevin Knight and Daniel Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1): 91–107, 2002.

Konstantinos Koumpis and Steve Renals. Transcription and summarization of voicemail speech. In *Proceedings of International Conference of Spoken Languag Processing (ICSLP 2000)*, Beijing, China, 2000.

Derek Lam, Steven Rohall, Chris Schmandt, and Mia K. Stern. Exploiting e-mail structure to improve summarization. Technical Report TR2002-02, IBM Watson Research Center, 2002.

Mirella Lapata. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pages 545–552, Barcelona, Spain, 2003.

David Dolan Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1992)*, pages 37–50, Copenhagen, Denmark, 1999.

Jung-Min Lim, In-Su Kang, and Jong-Hyeok Lee. Multi-document summarization in cross-language text. In *Proceedings of the Fourth NTCIR Workshop on Research in Information Access Technologies: Information Retrieval, Question Answering and Summarization*, Tokyo, Japan, 2004.

Chin-Yew Lin. ROUGE: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, Barcelona, Spain, 2004.

Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2003)*, pages 71–78, Edmonton, Alberta, 2003.

Fernando López-Ostenero, Julio Gonzalo, and Felisa Verdejo. Noun phrases as building blocks for cross-language search assistance. *Information Processing and Management*, 41:549–568, 2005.

Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of Research Development*, 2(2):159–165, 1958.

Inderjeet Mani. Summarization evaluation: An overview. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL) Workshop on Automatic Summarization*, 2001.

Inderjeet Mani, Gary Klein, David House, Lynette Hirschman, Therese Firmin, and Beth Sundheim. SUMMAC: a text summarization evaluation. *Natural Language Engineering*, 8(1):43–68, 2002.

Interjeet Mani and Eric Bloedorn. Summarizing similarities and differences among related documents. *Information Retrieval*, 1(1):35–67, 1999.

Ingrid Mårdh. *Headlinese: On the Grammar of English Front Page Headlines.* Malmo, 1980.

Sameer Maskey and Julia Hirschberg. Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization. In *Proceedings of the 9th European Conference on Speech Communication and Technology (InterSpeech 2005)*, Lisbon, Portugal, 2005.

Sameer Raj Maskey and Julia Hirschberg. Automatic summarization of broadcast news using structural features. In *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, Geneva, Switzerland, 2003.

Eric Mays, Fred J. Damerau, and Robert L. Mercer. Context-based spelling correction. *Information Processing and Management*, 27(5):517–522, 1990.

Andrew McCallum, Andrés Corrada-Emmanuel, and Xuerui Wang. The author–recipient–topic model for topic and role discovery in social networks, with application to Enron and academic email. In *Proceedings of the Workshop on Link Analysis, Counterterrorism and Security at the 2005 SIAM International Conference on Data Mining*, Newport Beach, California, 2005.

Gabor Melli, Zhongmin Shi, Yang Wang, Yudong Liu, Annop Sarkar, and Fred Popowich. Description of SQUASH, the SFU question answering summary handler for the DUC-2006 summarization task. In *Proceedings of the 2006 Document Understanding Workshop*, New York, New York, 2006.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Five papers on Wordnet. Technical report, Cognitive Science Lab, Princeton University, 1993.

Scott Miller, Lance Ramshaw, Heidi Fox, and Ralph Weischedel. A novel use of statistical parsing to extract information from text. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2000)*, pages 226–233, Seattle, Washington, 2000.

Smaranda Muresan, Evelyne Tzoukermann, and Judith L. Klavans. Combining linguistic and machine learning techniques for email. In *Proceedings of the ACL/EACL 2001 Workshop on Computational Natural Language Learning (ConLL)*, pages 290–297, Toulouse, France, 2001.

Ani Nenkova and Amit Bagga. Facilitating email thread access by extractive summary generation. In *Proceedings of 2003 Recent Advances in Natural Language Processing Conference (RANLP 2003)*, Borovets, Bulgaria, 2003.

Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The Pyramid method. In *Proceedings of the 2004 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2004)*, Boston, Massachusetts, 2004.

Paula S. Newman and John C. Blitzer. Summarizing archived discussions: A beginning. In *Proceedings of the 2003 International Conference on Intelligent User Interfaces (IUI 2003)*, Miami, Florida, 2003.

Douglas W. Oard, Julio Gonzalo, Mark Sanderson, Fernando López-Ostenero, and Jianqiang Wang. Interactive cross-language document selection. *Information Retrieval*, 7:203–226, 2004.

William Ogden, James Cowie, Mark Davis, Eugene Ludovic, Hugo Molina-Salgado, and Hyopil Shin. Getting information from documents you cannot read: An interactive cross-language text retrieval and summarization system. In *Proceedings of SIGIR/DL Workshop on Multilingual Information Discovery and Access*, Berkeley, California, 1999.

Naoaki Okazaki, Yutaka Matsuo, and Mitsuru Ishizuka. Improving chronological sentence ordering by precedence relation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 750–756, Geneva, Switzerland, 2004.

Mari Ostendorf, Aravamudan Kannan, Steve Austin, Owen Kimball, Richard Schwartz, and Jan Robin Rohlicek. Integration of diverse recognition methodologies through reevaluation of n-best sentence hypotheses. In *Proceedings of the DARPA Speech and Natural Language Workshop*, Pacific Grove, California, 1991.

Paul Over and Walter Liggett. Introduction to DUC-2002: An intrinsic evaluation of generic news text summarization systems. 2002. URL `http://duc.nist.gov/pubs.html\#2002`.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics(ACL)*, Philadelphia, PA, 2002.

Rebecca J. Passonneau and Ani Nenkova. Evaluating content selection in human- or machine-generated summaries: The Pyramid method. Technical report, Columbia, New York, NY, 2003. CUCS-025-03.

Martin Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

Michael J.D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7: 155–162, 1965.

Dragomir Radev, Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Çelebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, and Zhu Zhang. MEAD—a platform for multidocument multilingual text summarization. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*, Lisbon, Portugal, 2004.

Owen Rambow, Lokesh Shrestha, John Chen, and Christy Laurdisen. Summarizing email threads. In *Proceedings of the 2004 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2004) — Short Papers*, Boston, Massachusetts, 2004.

Edmund Rooney and Oliver Witte. *Copy Editing for Professionals.* Stipes Publishing Co., 2000.

Gerald Salton. *Automatic Text Processing.* Addison-Wesley Publishing Company, 1988.

Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, United Kingdom, 1994.

Richard Schwartz, Steve Austin, Francis Kubala, John Makhoul, Long Nguyen, Paul Placeway, and George Zavaliagkos. New uses for the n-best sentence hypotheses within the byblos speech recognition system. In *Proceedings of the IEEE International Conference onf Acousitcs, Speech, and Signal Processing*, San Francisco, California, 1992.

Richard Schwartz, Toru Imai, Francis Kubala, Long Nguyen, and John Makhoul. A maximum likelihood model for topic classification of broadcast news. In *Proceedings of the Fifth European Speech Communication Association Conference on Speech Communication and Technology (Eurospeech-97)*, Rhodes, Greece, 1997.

Siwei Shen, Dragomir R. Radev, Agam patel, and Güneş Erkan". Adding syntax to dynamic programming for aligning comparable texts for the generation of paraphrases. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, Syndey, Australia, 2006.

Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. Syntactic simplification for improving content selection in multi-document summarization. In *20th International Conference on Computational Linguistics (COLING2004)*, Geneva, Switzerland, 2004.

Sreenivasa Sista, Richard Schwartz, Timothy R. Leek, and John Makhoul. An algorithm for unsupervised topic discovery from broadcast news stories. In *Proceedings of the 2002 Human Language Technology Conference (HLT)*, pages 99–103, San Diego, California, 2002.

Jenine Turner and Eugene Charniak. Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 290–297, Ann Arbor, Michigan, 2005.

Lucy Vanderwende, Hisami Suzuki, and Chris Brockett. Microsoft Research at DUC2006: Task-focused summarization with sentence simplification and lexical expansion. In *Proceedings of the 2006 Document Understanding Conference (DUC 2006) at HLT/NAACL 2006*, pages 70–77, New York, New York, 2006.

Andrew J. Viterbi. Error bounds for convolution codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13:260–269, 1967.

Stephen Wan and Kathleen McKeown. Generating overview summaries of ongoing email thread discussions. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva, Switzerland, 2004.

Ruichao Wang, Nicola Stokes, William Doran, Eamonn Newman, Joe Carthy, and John Dunnion. Comparing Topiary-style approaches to headline generation. In *Lecture Notes in Computer Science: Advances in Information Retrieval: 27th European Conference on IR Research (ECIR 2005)*, volume 3408, Santiago de Compostela, Spain, 2005. Springer Berlin / Heidelberg.

Mark Wasson. Using leading text for news summaries: Evaluation results and implications for commercial summarization applications. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, Volume 2*, pages 1364–1368, Montreal, Quebec, Canada, 1998.

David M. Zajic, Bonnie J. Dorr, and Richard Schwartz. BBN/UMD at DUC-2004: Topiary. In *Proceedings of the 2004 Document Understanding Conference (DUC 2004) at NLT/NAACL 2004*, pages 112–119, Boston, Massachusetts, 2004.

Klaus Zechner. Automatic summarization of open-domain multiparty dialogues in diverse genres. *Computational Linguistics*, 28(4):447–485, 2002.

Liang Zhou and Eduard Hovy. Headline summarization at ISI. In *Proceedings of the HLT-NAACL 2003 Text Summarization Workshop and Document Understanding Conference (DUC 2003)*, pages 174–178, Edmonton, Alberta, 2003.

Liang Zhou and Eduard Hovy. On the summarization of dynamically introduced information: Online discussions and blogs. In *Proceedings of AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, Stanford, California, 2006.