Abstract

Effects of Topic Structure on Automatic Summarization

Natalie Margaret Schrimpf

2018

Automatic summarization involves finding the most important information in a text in order to create a reduced version of that text that conveys the same meaning as the original. In this dissertation, I present a method for using topic information to influence which content is selected for a summary.

This dissertation addresses questions such as how to represent the meaning of a document for automatic tasks. For tasks such as automatic summarization, there is a tradeoff between using sophisticated linguistic methods and using methods that can easily and efficiently be used by automatic systems. This research seeks to find a balance between these two goals by using linguistically-motivated methods that can be used to improve automatic summarization performance. Another question addressed in this work is the balance between summary coverage and length. A summary must be long enough to convey the information from the original text but short enough to be useful in place of the original document. This dissertation explores the use of topics to increase coverage while reducing redundancy.

There are several issues that affect summary quality. These include information coverage, redundancy, and coherence. This dissertation focuses on achieving coverage of all distinct concepts in a text by incorporating topic structure. During the summarization process, emphasis is placed on including information from all topics in order to produce summaries that cover the range of information present in the original documents. In this

work, several notions of what constitutes a topic are explored, with particular focus on defining topics using information from Rhetorical Structure Theory (Mann and Thompson 1988). The results of incorporating topics into a summarization system show that topic structure improves automatic summarization performance.

The contributions of this dissertation include demonstrating that focusing on coverage of the different topics in a text improves summaries, and topic structure is an effective way to achieve this coverage. This research also shows the effectiveness of a simple modular method for incorporating topics into summarization that allows for comparison of different notions of topic and summarization techniques.

Effects of Topic Structure on Automatic Summarization

A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by
Natalie Margaret Schrimpf

Dissertation Director: Robert Frank

May 2018

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank the Yale Graduate School for their generous funding that allowed me to pursue my studies. In particular, I am grateful for the A. Richard Diebold Jr. Graduate Fellowship and the Hall-Mercer Graduate Fellowship.

All of my committee members, Bob Frank, Drago Radev, and Owen Rambow, provided valuable feedback and pushed my research further. In particular, I would like to thank my advisor, Bob Frank. Through our many meetings over the years, whether in person or over Skype, he encouraged me to dive in and implement my ideas while also asking questions that pushed me to think harder and consider new approaches. In addition to my committee, I would also like to acknowledge the other faculty in the Yale Linguistics Department who provided valuable knowledge and support throughout my time in the program, in particular Gaja Jarosz who served as my advisor for my first two years. I am grateful to my fellow students who worked with me on homework, attended my presentations, and spent hours in the grad room debating, commiserating, and telling stories.

I would not be at this point without the professors of the Linguistics Program at Dartmouth College, including Jim Stanford, Dave Peterson, and Tim Pulju, who first introduced me to linguistics. I thank them for encouraging my curiosity about language.

Most of all I would like to thank the family and friends who have supported me during this process. My parents have been a constant source of support, advice, and love. I am extremely grateful to them for their encouragement and for instilling in me a love of language and learning, all the way back to those first readings of *Fox in Socks.* I thank

Matt, Carmen, and Alexandra for their support, advice, and cute pictures that provided a welcome distraction from the challenges of writing. I thank Mike for supporting me through the ups and downs of this journey. He helped me along the way with much-needed breaks, hugs, pizza, and love, among many other things. He told me I could do it, and as he likes to say, he knows these things.

To all of my family and friends who have provided a listening ear or an encouraging word over the years, thank you.

Chapter 1

Introduction

Summarization is the task of creating a shortened version of an input document that retains the important information from the original text but in a more concise form. The goal of summarization is to convey the main concepts of the original document so that a summary user can understand what the document is about without reading the entire text. With large amounts of text available online, it has become increasingly necessary to find ways to allow people to quickly and easily find the information they need. Summarization is useful for this task because it condenses information into a shorter form that can be read instead of a longer text if it provides all of the information a user needs or it can be read in order to determine whether the original text contains information relevant to the user's needs, allowing the user to decide which texts would be most useful. In order for summaries to achieve this goal, they must convey the important concepts from the text without including unnecessary information.

Different types of summarization systems, including extractive, abstractive, single-document, and multi-document, approach the problem in different ways. Although all of these systems have the same basic goal, there are advantages and challenges associated with all of these summarization types. Determining important information, selecting or generating informative but not redundant or extraneous sentences, and

combining sentences into a concise and coherent summary are all parts of a successful summarization system.

The work in this dissertation focuses on extractive single-document summarization. This involves selecting sentences from a document to produce a summary. In this work, several questions are explored, including what an ideal summary looks like and how to incorporate linguistic knowledge such as topic structure and rhetorical information into an automatic summarization system. The rest of this chapter provides more detail about the questions under consideration and suggests some of the approaches that will be used to explore these questions in the rest of the paper.

1 Overview of Summarization

1.1 Summary Examples

When exploring the question of how to perform automatic summarization, it is worth considering how people process text and write summaries as automatic summarization seeks to create summaries that are as useful and coherent as the ones written by people but without requiring human time and effort. The following example of a text, specifically a Wall Street Journal article, and a human-written summary of the text illustrate some of the strategies people use to select information and write summaries.

Example Text:

> RJR Nabisco Inc. is disbanding its division responsible for buying network advertising time, just a month after moving 11 of the group's 14 employees to New York from Atlanta.
>
> A spokesman for the New York-based food and tobacco giant, taken private earlier this year in a $25 billion leveraged buy-out by Kohlberg Kravis Roberts & Co., confirmed that it is shutting down the RJR Nabisco Broadcast unit, and dismissing its 14 employees, in a move to save money.
> The spokesman said RJR is discussing its network-buying plans with its two main advertising firms, FCB/Leber Katz and McCann Erickson.

"We found with the size of our media purchases that an ad agency could do just as good a job at significantly lower cost," said the spokesman, who declined to specify how much RJR spends on network television time.
An executive close to the company said RJR is spending about $140 million on network television time this year, down from roughly $200 million last year.

The spokesman said the broadcast unit will be disbanded Dec. 1, and the move won't affect RJR's print, radio and spot-television buying practices.

The broadcast group had been based in New York until a year ago, when RJR's previous management moved it to Atlanta, the company's headquarters before this summer.
One employee with the group said RJR moved 11 employees of the group back to New York in September because "there was supposed to be a future." He said the company hired three more buyers for the unit within the past two weeks, wooing them from jobs with advertising agencies.

The RJR spokesman said the company moved the 11 employees to New York last month because the group had then been in the midst of purchasing ad time for the networks' upcoming season. "The studies {on closing the unit} couldn't be completed until now," he said.

The group's president, Peter Chrisanthopoulos, wasn't in his office Friday afternoon to comment.

Example Summary:

In a cost-cutting move, RJR Nabisco will shut down its broadcast division and dismiss the unit's fourteen employees, some of whom had just been sent back to New York from Atlanta. The broadcast unit was responsible for buying network advertising time.

A company spokesman who confirmed the shut down said that, based on the size of the company's media purchases, its two main advertising firms could provide the same service at a lower cost.

The same spokesman also explained that studies on closing the unit were not completed until after moving eleven of the unit's employees from Atlanta to New York and hiring three more buyers within the past two weeks.

The move is not expected to affect the company's print, radio and spot television buying practices.

The first paragraph of the summary is roughly the same as the first paragraph of the text, but the information is combined into sentences differently. Both introduce the main idea of the text, which is that Nabisco is closing its broadcast division. The second

paragraph of the summary describes the highlights of the second and third paragraphs in the text, which discuss the motivations for the closure. Until this point, the order of information in the summary corresponded to the order in the original text. However, the third and fourth paragraphs of the summary present information in a different order. The fourth summary paragraph relates to information from the middle of the text, and the third paragraph is related to the end of the text.

This summary demonstrates some of the important aspects of summary creation. While there is some overlap in the use of particular words and phrases, there are also differences in the way information is presented. In some cases, information from multiple sentences is combined into a single sentence, and in other cases part of a sentence in the original text is presented as its own sentence. There are also word differences. While the original text refers to "disbanding" the division, the summary uses the phrase "shut down." The event is described as a "move to save money" in the original and as a "cost-cutting move" in the summary. Although these phrases have the same meaning, the difference is relevant when thinking about automatic summarization. People reading a text are able to understand its meaning and condense it to the most important information. With their understanding of the information, they can extract words, phrases, or sentences for a summary while also generating new sentences or using related words to describe the same concept. These relatively easy tasks for people are more challenging for an automatic system. Generating coherent and grammatical sentences is particularly difficult, so most automatic summarizers avoid the problem by extracting complete sentences from the original text. Similarly, combining sentences or shortening sentences requires understanding the roles played by each piece of the text so that these roles are

maintained in the summarized version. Using related words that do not appear in the original text requires access to a thesaurus or other resources that allow for more abstract representations of meaning and understanding relationships between words.

Beyond the task of actually extracting or generating the language for a summary, automatic summarization also involves determining which information should be included in the summary in the first place. Because a summary is shorter than the original text, it obviously does not contain all of the same information. For instance, in the example above, the information about the company becoming private as well as the direct quotes are not included in the summary. This illustrates one of the most important parts of summary creation, which is deciding which information to include. Judging importance is a complex task with no straightforward solution as people disagree on importance and there is not a single correct answer. This point is illustrated below.

1.2 Summary Variation

When comparing different summaries of the same text, there is clearly variation between summaries in terms of which specific information is included and how that information is expressed. The following examples are two summaries of the same news article, written by different people (DUC 2002).

Summary 1:

> On Wednesday, Hurricane Gilbert, a category 5 storm, the strongest and deadliest type, slammed into the Yucatan Peninsula with 160mph winds causing heavy damage to the resort areas of Cancun and Cozumel. More than 120,000 people on the northeast Yucatan coast were evacuated. Earlier the storm killed 19 people in Jamaica and 5 in the Dominican Republic. It destroyed an estimated 100,000 of Jamaica's 500,000 homes. Mexican officials were often helpless as floods closed roads and downed power lines disrupted phone service. The already record-setting storm is expected to intensify as it leaves the Yucatan and again moves over water.

Summary 2:

> Hurricane Gilbert slammed into Mexico's Yucatan Peninsula on Wednesday. The eye hit Cancun and Cozumel at 8 a.m. carrying winds up to 160 mph. Heavy flooding, destroyed slums, and severed communications and power lines were reported. No relief efforts were yet underway and bands of youth were said to be looting in Cancun. About 175,00 people had been evacuated from the coast. Gilbert, a Category 5--the deadliest--has already killed 24 people in the Caribbean. It is now moving over the Yucatan toward the Gulf of Mexico where warnings have been issued and ports and airports closed.

These summaries convey roughly the same information from the original text although there is variation in how the information is expressed, including the words used and how information is organized into sentences. For example, the first sentence of both summaries includes the information that Hurricane Gilbert hit the Yucatan Peninsula on Wednesday. However, the first summary also includes information about the wind speeds and category of the storm in that sentence, while the second summary includes those pieces of information in other parts of the summary. Those details, among others, are included in both summaries, but some details such as how many homes were destroyed and airport closings are included in only one of the summaries. This indicates that there is some variation in what summaries include, but there is clearly a lot of overlap between different summaries of the same text. Similar words and phrases are used, and importantly similar information is conveyed. This suggests that documents contain a main theme or certain crucial information that must be included in a summary for that summary to be a good representation of the original text. As long as that information is conveyed, specific details of the words and phrases used and how information is organized into sentences is less important. What is most important is the information

coverage of a summary, and automatic summarization should therefore focus on achieving this coverage.

In general, while there is variation between different summaries of the same text, there is a sense in which different summaries convey the same main argument. Different pieces of information from the original text can be combined into distinct summaries that all capture the theme of the text. This fact indicates that what qualifies as a summary, and in particular a good summary, goes beyond the inclusion of particular words, phrases, or sentences from the original text. The quality of a summary should be considered at a more abstract level, just as a text is not simply the sentences it contains but the meaning and argument created by these sentences and their combination.

1.3 Idea of an Optimal Summary

The question of what differentiates good summaries from bad summaries relates to the goal of the summary. Why is a summary created and what purpose will it serve for its user? For practical purposes a summary may be more or less useful depending on whether the user simply wants a general sense of the original text or whether they want to read the summary in place of the original. It is also possible that a user wants a summary that describes the issue from a particular perspective or wants to know the information from the original text that relates to a certain topic or question. In these cases, the quality and usefulness of a summary will depend on how well it meets these specific requirements. In the more general case without restrictions on the summary, the problem of determining the quality of a particular summary and differentiating between summaries is more challenging as well as more interesting. In general, what characteristics should a summary have? What are the common properties shared by

different summaries of the same text? Since there is variation among different summaries, there is clearly not an exact formula for determining what a summary should contain. However, the similarities found between summaries indicate that choices about what to include in a summary are not completely random.

In general, an optimal summary should convey the most content from the original text given constraints such as length. It should contain broad coverage of the information in the text. For example, if a text has five main topics, it should prefer limited coverage of all topics rather than in-depth coverage of only one topic. Ideas of what constitutes a topic will be discussed in later chapters. This concept of what is optimal is based on the intuition that summaries never include all of the information from the original text, so it is better for the information that is left out to be the more in-depth and detailed parts of the text rather than more general concepts. For example, in a summary of a research article, including some information about the hypothesis, methods, and results will produce a more informative summary than one that includes a detailed description of the methods but no other information. Wider coverage allows a summary reader to have an idea of all of the concepts in the text, and if more specific information about a concept is desired, the reader has an idea that it is contained in the original text. If more general concepts are left out of a summary in favor of details about a single concept, a reader has no way of knowing that information about the other concepts is contained in the original text. An optimal summary should leave a reader with enough information to reconstruct the main ideas of the original text.

In general, texts can be thought of as containing a single main argument or idea, with the information in the text contributing to the argument in different ways, a theory

which has been formalized by Rhetorical Structure Theory (Mann and Thompson 1988). A summary should convey this same argument. In order for the input text and its summary to convey the same argument, we must assume that the same argument can be conveyed by different amounts of information. While the original text includes, either explicitly or implicitly, all of the information necessary to understand the main idea, the summary must leave out certain information. However, there are restrictions on which information can be left out and not included in the summary. In particular, information that is not included in the summary should be inferable from the summary. For example, in a text that describes an experiment, there will be specific details of how the experiment was conducted, such as the materials that were used or the number of participants. While a summary does not need to include all of these details, the summary must include the fact that an experiment was conducted. With the information that an experiment was conducted, readers can infer that there were some number of participants but without that information those details cannot be inferred or reconstructed. Therefore, a summary must contain enough information so that the details that are not included can be inferred.

The idea that texts contain a central argument that relates different pieces of the text to each other is useful for understanding what a good summary is and understanding the variation between different summaries of the same text. One way to define what constitutes a good summary is that a good summary is one that conveys the main argument of the original text. Different parts of the text contribute to the argument in different ways. Some parts of the text may be more central to the argument than others, and some parts can contribute very similar information. The challenge when creating a summary is to determine the argument and reconstruct the argument in a shortened form.

This process involves not only selecting relevant pieces of information from the text but also synthesizing these pieces of information in a way that preserves the original argument. At a high level, a text and a summary of that text should convey the same information. At lower levels, there are many differences between a text and a summary, including the amount of detail included, and at an even lower level the words that are used. The number of differences between a text and a summary as well as the amount of variation between different summaries of the same text depends on the level of granularity being considered. There is more variation at lower levels, such as the words, phrases, or sentences used. At higher levels of consideration, such as the main topics or themes, there is less variation. Therefore, at a high level, it is more straightforward to understand what constitutes a good summary. A summary should convey the same themes and arguments as the original text. However, the challenge is to find a principled way to determine the main argument and judge different parts of the text in terms of their relation to this argument. On the other hand, the situation is different when considering summaries at a low level. Looking narrowly at particular words or sentences does not provide enough information about the quality of a summary as a whole, but it is easy to represent a text simply as the words or sentences it contains. The tasks of creating and evaluating summaries present these opposing challenges of accurate representation of information and ease of representation.

While a summary should convey the same argument as the original text, a summary has more limited space to form the argument. It is important to determine which information is most central to the argument and also combine information into a coherent summary. Many automatic summarization systems simply select a set of

sentences from the input and present them as a summary. Sentences are scored and selected independently. This independence does not capture the fact that what is important is that the collection of sentences together conveys the argument. If two sentences are similar and contribute the same information to the argument, a good summary should likely contain only one of these sentences to ensure that other parts of the text are represented in the summary. All sentences in the summary should relate to the argument, but they should not be too similar to each other. In order for a summary to contain maximum coverage of the ideas and concepts of the text, no concept should be overrepresented. If selecting two sentences, choosing two very distinct sentences will likely result in a better summary in terms of maximum coverage than selecting two very similar sentences. Choosing sentences for a summary independently does not enforce this restriction. In addition, different combinations of sentences may convey the text's argument equally well. The quality of a summary does not depend simply on the quality of individual sentences but on their combination and the meaning they convey when taken together. Choosing and evaluating sentences independently does not provide a way to check that the main argument of the text is conveyed, and therefore an important part of what determines the quality of a summary is ignored.

One way to approach the problem of determining whether a summary conveys the same argument as the text it is based on is to break the argument down into smaller pieces. Every smaller unit in the text should be related to the argument in some way. In that case, it would be possible to state that a summary conveys the argument if it conveys each piece of the argument from each relevant unit of the text. The argument can be thought of as having several components that together form the complete argument. A

summary can be judged on whether it contains each of these components. It is necessary to determine the main idea of each section, but once those have been determined they provide a way to decide whether the overall argument is conveyed. Breaking the argument down in this way allows for more consideration of whether the combination of information in the summary is good and not simply whether individual pieces of information are relevant.

2 Research Questions

2.1 Representing Meaning

Given the need to determine which information is important and should be included in a summary, one question is how to represent the information content of a text. This includes the questions of what kinds of text and meaning representations capture information relevant to summarization and which representations are easily available for use on this task.

It is difficult to determine automatically what the meaning of a text is. Meaning is based on many factors. The meaning of larger sections of text depends on the meanings of its parts as well as how they are connected. The meaning of a document depends on its sentences and relationships between those sentences, including sentence order, repetition of entities, and discourse relationships that describe how one sentence is connected to another, such as through a cause and effect relationship. The meaning of a sentence depends on the words it contains, their order and syntactic relationships, as well as the context of the sentence.

Determining meaning is a hierarchical process with the meaning of higher levels, such as documents or corpora, dependent on the meaning of lower levels, such as words

and sentences. However, representing meaning at different levels is not an entirely straightforward task. In particular, how to determine and represent meaning automatically are challenging questions. While a person may have no trouble reading and understanding a text, that same text could be difficult for an automatic system to process and extract meaning that can be used for a given task, such as summarization. Even simple sentences can present challenges. For example, "The cat did not eat the food" contains several elements that can be difficult to process. To correctly capture the meaning of the sentence, an automatic system must be able to understand negation, the difference between present and past tense, the distinction between subject and object, and the anaphoric reference of "the cat" and "the food." This requires a semantic representation of the sentence. Simply looking at the words in the sentence does not necessarily lead to the correct interpretation as that would eliminate the information from word order and syntax.

The challenges increase as the tasks of understanding and representing meaning scale to larger pieces of language, such as paragraphs or documents. Creating a clear and concise representation of the meaning of a document is not an easy task. It relies on understanding the meanings of individual sentences as well as their interactions and relations to each other. Similar to the challenges discussed above for understanding sentence meaning, understanding document meaning involves challenges such as determining the antecedent of a pronoun when they appear in different sentences or understanding how discourse markers signal particular relationships between sentences. It is particularly challenging to determine document meaning automatically, without the influence of human judgment and interpretation. Therefore, in many natural language

processing tasks, simplifications and heuristics are used. One of the most common simplifications is the "bag of words" assumption, which reduces a piece of text, such as a sentence, to an unordered list of the words it contains. While this strategy does not solve the problem of representing meaning, this is a straightforward way to simplify the task of representing meaning as it requires no special processing or annotation. An example at the other end of the spectrum from representing meaning using a bag of words would be to represent texts in terms of their entailments. In that case, the meaning of a text is whatever is entailed by that text. Two texts could easily be judged to be similar if they shared the same entailments. However, in contrast to the simplicity of producing a bag of words representation, determining and representing entailments requires much more effort. For example, the entailments of a particular piece of text, such as a sentence, can differ depending on context. Recognizing textual entailment (RTE) is its own research area that seeks to solve the problem of automatically determining which information logically follows from a text (Dagan et al. 2006). Because determining entailments presents its own challenges, using this information as a representation of meaning to be used in another task like summarization only increases the effort and resources needed to perform summarization.

These examples raise the question of how to balance the opposing goals of simple, efficient processing and accurate representations of the data. If representing the data becomes too cumbersome, then automatic systems will be less able to operate given time and processing constraints. On the other hand, if the data representation is too simple, it may not be informative enough to be used for complex language processing tasks. This relates to the larger question of whether to use more sophisticated linguistic

information in order to emulate how humans perform a task or whether to use whichever methods achieve the best performance when evaluated on that task. The research in this dissertation seeks to incorporate both of these goals by finding linguistically-motivated methods that also improve automatic summarization performance. Therefore, it focuses on exploring linguistic information, including topic structure and rhetorical structure, and finding ways to make this information accessible to an automatic system. Therefore, compromises must be made between using an ideal representation of meaning and one that is more practical and accessible.

2.2 Summary Length and Maximum Coverage

Another question related to summarization is how long a summary should be. Summaries must be short enough to meet any given length restrictions and to be useful in place of a full document. On the other hand, summaries must be long enough to convey the important information from the original text. This raises the question of what the ideal length for a summary should be and if it is possible to determine that length automatically based on properties of the text. Related to the idea discussed above that different summaries seem to convey the same information even if they use different words, phrases, and sentences, the idea of an ideal length for a summary of a given text assumes that a text contains some amount of crucial information and a summary should contain all of this information, but with no superfluous information and no redundancy. If this crucial information can be determined, the task of summarization becomes simply deciding how to express this information. However, determining which information is required amounts to the other component of summarization, which is selecting what to include. It would also involve the added difficulty of going beyond ranking information

and selecting a specified amount to the more challenging task of deciding how much information to select. As there are no clear units to use for representing information content, and the same information can be expressed in multiple ways, determining the appropriate length for a summary is therefore related to the issue of how to represent the content of a document. The issue of the ideal summary length will be discussed further as it relates to the connection between summarization and basic text compression. The question of representing the content of a document and ensuring that a summary contains the most coverage with the least redundancy is one of the main questions that will be explored in this dissertation.

3 Use of Topics

As discussed above, one of the most important parts of summarization is deciding which information to select for a summary in order to convey the meaning of the original text. When discussing the qualities of a good summary, it was suggested that one way to determine whether a summary conveys the same argument as the original text is to break the argument down into smaller pieces and check whether these smaller components are represented in the summary. This idea can also be applied to summary creation. To produce a summary that captures the same information as the original text, a text can first be divided into smaller cohesive sections. Then the summarization process can focus on representing each of these sections in the final summary. To accomplish this task, the work in this dissertation uses topic structure. Texts contain different topics or groups of sentences that are more related to each other than they are to sentences in other groups. At a high level, texts can be explicitly divided into topics. For example, scientific papers discussing experiments can be divided into sections such as the hypothesis, methods, and

results. At a different level of granularity, texts seem to be organized around smaller topics that represent what different sentences have in common. The following three example texts illustrate how different types of texts can be organized into topics.

3.1 Topic Examples

The first text includes excerpts from a scientific article describing an experiment. The topic headings are in bold, and sub-topic headings are italicized.

Text 1 (Hyona et al. 2002):

> **Overview of the Experiment:**
> The eye movements of college students were recorded as they read two multiple topic, expository texts for the purpose of summarizing each text from memory. The frequency and duration of fixations were classified into four categories for each sentence in each text…
> **Method:**
> *Participants*
> Participants were 48 students (29 women; age range: 20–36 years) enrolled at the University of Turku, Finland…
> *Materials*
> Two multiple-topic expository texts were used as stimuli, the Energy and Endangered Species texts…
> **Results:**
> *Data Reduction*
> Eight measures of eye movements were computed for each sentence in both texts…

This example illustrates how texts can be divided into topics using explicit divisions and labels. The separation between topics indicates a change in information, and the labels describe the topic of each new section. This example also demonstrates that topics can exist at different levels of granularity, with sub-topics such as *Participants* and *Materials* occurring within the larger topic of *Method*. Because documents such as this one contain explicit labels, the task of dividing a text into topics is straightforward.

The second example presents a different type of text, a movie review. This text contains no explicit labels or divisions, but the reader can determine that several topics are discussed.

Text 2 (Pang and Lee 2004):

> **In "Magic Town," Jimmy Stewart is in peak form playing a pollster who heads to a "perfect" town to gauge their reactions on the sorts of issues that only poll-takers care about. However, as time progresses, and he finds himself falling in love with the town (and a woman), he begins to see that what he's doing is wrong. The plot is standard stuff, but that's not important.** *In a film like this, it's the caliber of the actors that make or break it. Obviously, since Stewart stars, that's practically a non-issue. He's great in the film, as usual. There's just something about him that always manages to be endearing, even when he's deceiving the town folk. You're always on his side, and you desperately want to see all of his goals come to fruition.* The film was directed by a longtime Frank Capra script-writer, and it shows. This is the type of feel-good picture that Capra is famous for. By the time "the end" shows up on screen, everything has been wrapped up very nicely. There are no loose ends, and virtually every character gets a happy ending (those that deserve one, anyway). In this age of cynicism, it's refreshing to see a movie so upbeat.
> "Magic Town" is a delightfully entertaining motion picture. If you believe all old movies are slow-paced, you'd be well-advised to check this one out.

Several topics are evident in the text. The text in bold discusses the plot of the movie. The italicized text describes the acting, and the underlined text is about the directing. With an understanding of what movies and movie reviews generally include, it is relatively simple for a human reader to determine as well as label the topics in this text. Although there are no labels, as in the previous text, movie reviews are similar to scientific articles in that there are certain types of topics that these texts typically discuss. For scientific papers, these include the methods and results, while for movie reviews, topics include the plot and the actors. Therefore, certain types of texts follow a sort of template with similarities in topics and structure regardless of differences in content. Although there are clearly multiple topics discussed in the movie review example,

determining boundaries between topics or topic labels is more challenging for an automatic system. Without explicit divisions, other methods such as looking for word similarity or finding discourse relationships must be used to automatically divide a text into topics.

The third example illustrates another type of text, a news article. It contains several paragraphs from a Wall Street Journal article. As in the previous text, there are no explicit topic labels.

Text 3 (Carlson et al. 2002):

> Meanwhile, sterling slumped on news that the United Kingdom posted a wider-than-expected trade deficit in September. The news also knocked the British unit to below 2.95 marks in London, but a bout of short-covering helped sterling recoup some of its earlier losses.
>
> On the Commodity Exchange in New York, gold for current delivery jumped $3.20 to $370.20 an ounce. The close was the highest since Aug. 15. Estimated volume was a light two million ounces.
>
> In early trading in Hong Kong Wednesday, gold was quoted at $368.25 an ounce.

The first half of the text discusses the topic of sterling, while the second half discusses gold. In contrast to the previous two examples, this text and news articles in general do not follow a specific template or choose from a limited set of topics. The task of determining the topics and dividing the text accordingly is challenging without any guiding information. However, it is relatively simple for people to understand the topic of a section and notice when the topic changes and when it remains the same. For an automatic system, there are several elements of a text that could be useful for this purpose, including changes in the vocabulary, introduction of new entities, paragraph breaks, and the presence of discourse markers and relationships. Properties of the text

itself, such as these types of information, can be used to determine topics when no explicit information or human judgment is available.

3.2 Motivation

These three examples illustrate how different types of texts are organized around topics. The overall generalization is that texts can be divided into topics, possibly of different granularity levels, and these topics represent the main ideas contained in the text. In order to produce summaries that convey the same information as the original text, summaries should be created based on these topics, with an emphasis on covering the information from different topics in order to create the optimal coverage of the information in the text. Topics are a way to organize a text around the main ideas it expresses and are a way to capture important relationships between parts of a text in terms of the content they relate to. Conveying the same content as the original text is a crucial part of summarization and is therefore motivation for using topics to organize the text into groups of related content.

One crucial factor that motivates grouping texts into topics for summarization has to do with summary length. A summary is a condensed form of the original text, and summaries can vary in length. One of the challenges of summarization is determining how to convey the same information as the original text in a more limited space. In order to convey the same information, there should be an emphasis on covering the text by including some amount of information about all of the important ideas and by limiting redundancy and in-depth coverage of a particular topic in favor of wider coverage of all topics. How the information from the text is covered and the degree of coverage of any given topic depends on the summary length. A shorter summary will involve more

general coverage of topics, while a longer summary will allow for more detailed coverage.

3.3 Notions of Topic

As this work aims to use topics to inform summarization choices, an important consideration for comparing different notions of topic relates to how useful they would be for summarization. For example, summarization is generally used to condense sections of text larger than the sentence level, such as documents containing multiple paragraphs or pages. Therefore, a notion of topic that defines topics at the sentence level is less useful for summarization unless it can be extended to larger sections of text. Similarly, documents can be characterized by topics of different granularity, such as the sub-topics seen in the example above. If sub-topics become too narrow, including all topics in the limited space of a summary becomes challenging. As suggested by these examples, there is no single definition of topic in terms of topic content or size. This work seeks to explore a few notions of topic, with emphasis on defining topics in a way that can be applied to single-document summarization. The goal is to produce a summary of a single document with coverage of all important concepts in the document, with topic structure as the crucial element for ensuring this broad coverage.

This dissertation will explore the question of how to define topics. The linguistics and computational linguistics literature contain a variety of notions of what it means to be a topic. There are two related issues. The first issue is understanding theoretically what topics are and how people are able to process and determine topics. The other issue is how to define and use topics automatically. Given a notion of what a topic should be, how can that be applied to a set of texts? This work will explore both how topics can be

understood more abstractly and how the idea of topics can be applied to texts to create meaningful structure to be used for other tasks.

In particular, the question of how to use topics for automatic summarization will be explored. This introduction has suggested motivation for why the type of structure given by topics should be useful for summarization. This work will explore the specific details of how to incorporate topic information into a summarization system that produces extractive summaries. Experiments are performed in a modular fashion that allows for direct comparisons between the results when using topics and not using topics as well as between the use of different notions of topic.

4 Outline of the Dissertation

Chapter 2 describes different approaches to automatic summarization. It reviews previous research in the area from early work using relatively simple methods to recent work using more complex techniques. This chapter also draws a connection between summarization and text compression. It describes some related research on compression and explores how the methods used for compression relate to the task of summarization. As mentioned above, this dissertation focuses on the use of topic structure for summarization, and an important question to consider is how topics are defined. Chapter 3 presents topic definitions from the linguistics literature. It describes different notions of topic, with emphasis on two notions of topic that will be used in later experiments. Chapter 4 presents the experiments exploring the use of topics for automatic summarization. It describes the methods, including how summarization is performed and how the impact of topics is evaluated. It presents the results and includes discussion and

analysis. Chapter 5 concludes the dissertation by returning to the questions introduced in this chapter and highlighting the results and contributions of this work.

Chapter 2

Approaches to the Task of Summarization

1 Overview

There have been many approaches to the task of summarization in past research. Several types of summarization exist based on the goals for the summary and the methods used to create the summary. The first half of this chapter will give an overview of different types of summarization, including extractive and abstractive. It will also describe approaches that have been used in previous research, with specific examples of how summarization systems work. The second half of this chapter discusses another way to think about summarization, specifically as a task related to basic text compression. Previous work drawing a connection between compression and summarization will be discussed. This section also explores the type of information used by compression algorithms and the need for compression to be extended from finding surface-level similarities to finding similarity of meaning to be used for summarization. The chapter also includes the suggestion of an intermediate step of using topic structure as a way to create summaries that include the optimal coverage of the information in the original documents. This idea will be pursued in detail in the following chapters.

1.1 Types of Summarization

There are two main approaches to automatic summarization. The first main approach for creating a summary from an input text uses *extractive summarization*, and

the other uses *abstractive summarization*. Extractive summarization involves creating a summary by extracting complete sentences from the original document (Yih et al. 2007; Conroy et al. 2006; Wong et al. 2008; Christensen et al. 2013). In contrast, abstractive summarization involves generating a summary using the main ideas and concepts from the original text but without simply copying sentences (Moawad and Aref 2012; Genest and Lapalme 2012; Liu et al. 2015). Most research focuses on extractive summarization due to the added difficulty of abstractive summarization, including creating a representation of a text's meaning as well as generating language.

These different summarization strategies present different challenges. Extractive summarization results in summaries containing grammatical sentences. The challenge is to determine which sentences are important and contain the most relevant information in order to convey the meaning of the original text in a more limited space. In addition, extractive summarization involves the difficulty of creating a summary in which the selected sentences fit together when removed from their original context. On the other hand, the challenge of abstractive summarization is to generate well-formed sentences from a more abstract representation of the original document's meaning. In addition to determining the important concepts in a document, these concepts must be translated into grammatical natural language sentences. Because of this added difficulty of generating sentences, most research focuses on extractive summarization and methods to improve extracted summaries. Some of the challenges of abstractive summarization will be discussed further below.

Another main distinction in summarization methods is between generic and query-based summarization. Query-based summarization involves producing a summary

with information relevant to a specific question or topic. There may be several main concepts discussed in a single document, and a query-based summary focuses on one of these concepts to produce a summary that is informative with respect to that particular concept. On the other hand, in generic summarization the goal is to produce a more general summary that includes information about all of the important concepts from the original document. The main difference between these two methods is how information is selected. For generic summarization, it must be determined which information is important, while in query-based summarization, the query guides the selection of information for the summary. Overall, similar techniques can be used for both of these tasks.

A final distinction that is important for summarization is whether a summary is based on a single document or multiple documents. Single-document summarization involves creating a summary of one document, while multi-document summarization involves producing a summary of several related documents. In multi-document summarization, the input is a set of documents that share a common theme, such as a set of news articles that all describe a recent event. Overall, the strategies for finding important information are the same for single-document and multi-document summarization. However, in multi-document summarization there is more emphasis on determining similarity between sentences in order to select a single instance from a set of similar sentences. Because the documents describe the same idea, it is expected that there is a lot of overlap between sentences, particularly sentences from different documents. Therefore, the tasks of determining which sentences contain the same information and

reducing redundancy are even more important for multi-document summarization than for single-document summarization.

1.2 Issues in Summarization

There are several challenges for summarization that affect the approaches that are used. These issues include coverage, redundancy, and coherence. Coverage refers to how much information and which information from an original document is included in a summary. In general, a summary should convey the same information as the original document. However, a summary is restricted in length, so a main component of summarization is determining which information is covered in a summary. This involves balancing coverage and length. Increasing coverage typically increases summary length. While longer summaries may be more informative, they are also less useful as summaries if they are not a significant reduction from the original text. Therefore, automatic summarization methods seek to balance these two qualities in order to create summaries that are useful with respect to both content and length.

Redundancy is another issue related to coverage. As a summary is a shortened form of the original document, one way to achieve the necessary reduction is to limit redundancy in the summary. Including two sentences that contain the same information is not an efficient use of a summary's limited space. Many summarization methods attempt to determine the similarity between different pieces of a text in order to prevent including redundant information.

A final issue in summary creation is coherence. Well-formed documents generally follow certain organizational principles. Sentences are presented in a particular order so that later sentences can depend on previous sentences. For example, sentences providing

background information are presented first, and pronouns and other referring expressions are only used after their antecedents have been introduced. However, these constraints are not necessarily followed by automatically-produced summaries. Unless adjacent sentences are selected, summarization involves removing sentences or sections of text from their original context. Combining sentences from different parts of a document is an important part of summarization, but it can result in summaries that are not very coherent. Therefore, some automatic summarization systems specifically focus on improving summary coherence.

These issues of coverage, redundancy, and coherence affect summary quality and are important to consider when determining how to create summaries automatically. The approach proposed in this dissertation is based on the goal of creating a summary with the optimal coverage of the original document. In practice, determining the amount of distinct content and the appropriate amount of coverage in a summary are challenging tasks. Although it is difficult to determine the ideal length for a summary, the goal of summarization should be to create the best summary given the length constraints. For example, one major task of summarization is to remove redundancy so that what is left is the set of distinct information in the text. This means that summaries should include coverage of all distinct concepts in the text while limiting redundant coverage. One way to accomplish this is to conceptualize texts as composed of several topics, each of which should be included in the summary.

Organizing texts in terms of topics provides a way to capture this process of optimizing the summary content given the summary length. Texts can be separated into topics, and then sentences can be chosen to represent each topic. This ensures coverage of

different ideas. For example, if a text contains two topics, in order to produce the optimal coverage of the information from the text, sentences from both topics should be included in a summary. When a summary is allowed to contain only a few sentences, one sentence can be chosen from each topic rather than choosing multiple sentences from the same topic. As the number of sentences increases, a balance of sentences from the topics should be chosen. Topics provide a way to organize the information in a text so that sentences can be chosen in a way that produces wide coverage of the subjects of the text and reduces redundancy by preventing disproportionate coverage of any one topic. Wide coverage and reduced redundancy are both important qualities for a summary, and therefore there is strong motivation for using the topic-based optimization method. The next chapter delves into the issues of different notions of what it means to be a topic and how to divide texts into topics. Using topics for summarization is an important part of the current work and is at the center of the experiments described in chapter 4.

The rest of this chapter describes previous research and the methods that have been used to tackle the summarization challenges of coverage, redundancy, and coherence.

2 Methods for Summarization

Since extractive summarization involves choosing sentences from the original text and combining them into a summary, one of the most important parts of an extractive summarization system is how sentences are selected. Many people have proposed systems that use graph-based methods for summarization, choosing sentences based on their connections in the graph (Erkan and Radev 2004; Mihalcea and Tarau 2004; Christensen et al. 2013). Other approaches assign scores to words and sentences based on

features such as how frequent a word is or how often a word occurs at the beginning of a document (Yih et al. 2007; Conroy et al. 2006). Sentences are chosen to be included in a summary based on these scores.

2.1 Sentence Scoring

Given that there are many ways to score sentences, Ferreira et al. (2013) describe and compare a variety of sentence scoring methods that are commonly used for choosing sentences for extractive summarization. The first group of scoring techniques considered by Ferreira et al. includes word-based methods. Word frequency is a basic method that involves selecting sentences for a summary based on the frequency of the words contained in a sentence (Luhn 1958; Lloret and Palomar 2009; Gupta et al. 2011). The idea of using word frequency as a measure of importance appeared in some of the earliest work on automatic summarization (Luhn 1958). The intuition of this method is that more frequent words are more important than less frequent words because words that occur often are more likely to be related to the subject of the text. In this method, word frequencies within the document to be summarized are counted and summed over all words in a sentence, as shown below.

(2.1) $frequency_{doc}(word) = count_{doc}(word)$

(2.2) $Sentence\ score = \sum_{i}^{length} frequency_{doc}(w_i)$

One issue with this method is that some frequent words are not very informative. For example, function words such as *the, a,* and *is,* are likely to be very frequent throughout a document, but they have little influence on the content of a sentence and are not good indicators of sentence importance.

Because of the problems with looking at frequency alone, a variation on the frequency method is to consider term frequency and inverse document frequency (TFIDF) (Robertson 2004; Sparck Jones 1972; Tokunaga and Makoto 1994; Murdock 2006). This technique has been used in the context of document retrieval to distinguish relevant documents from a larger collection of documents and has also been used in the context of summarization to determine which sentences contain important information. This method considers how frequently words in a sentence occur relative to their overall frequency in the document.

(2.3) $$term\ frequency_{sentence}(term) = count_{sentence}(term)$$

(2.4) $$inverse\ document\ frequency = log \frac{\#\ of\ sentences}{\#\ of\ sentences\ containing\ term}$$

(2.5) $$tfidf = term\ frequency_{sentence} \times inverse\ document\ frequency$$

The frequency of a word within a sentence is counted for term frequency. Inverse document frequency is calculated by taking the log of the number of sentences in the document divided by the number of sentences that contain the word. The log is used to decrease the effects of very frequent words. For example, thinking on a larger scale than a single sentence, the difference between 1 million and 2 million occurrences of a word is not necessarily significant as both numbers represent very high frequency. Taking the log of the ratio prevents importance from increasing proportionally, which could result in frequent words being given too much importance. Similar to the basic frequency method, the score for a sentence can be calculated by summing over these weighted frequencies of the words in the sentence.

With this method, words that appear frequently within a sentence but not frequently throughout the entire document are considered more informative than words

that are frequent throughout the entire document, since those words do not allow for differentiation between sentences in the document. For example, a sentence with several occurrences of the word *the* will not be scored highly simply due to the frequency of that word, since it will be frequent throughout the entire document. Sentences have a higher score if they contain more informative words rather than simply the most frequent words.

Several scoring methods consider properties at the sentence level as opposed to word level properties. One example is sentence position (Gupta et al. 2011; Abuobieda et al. 2012). Several methods consider sentences at the beginning of a document or paragraph more important than other sentences based on the intuition that the most important information is presented first and that the first sentences are crucial for understanding the information that follows. Another method is to base a sentence's score on the presence of certain types of elements including the number of cue words and phrases, such as "in conclusion" and "the most important", contained in the sentence or whether a sentence contains numbers such as dates or percentages.

All of these methods produce scores for sentences. Using these scores, sentences can be ranked for importance. To produce a summary of length n, the n highest-ranked sentences according to these methods can be selected for inclusion in a summary. These types of methods mostly focus on the coverage aspect of summary quality by emphasizing the task of determining and selecting the most important or salient information.

Yang et al. (2017) demonstrate how word and sentence features can be used to rank sentences and produce a summary. They train a SVM classifier to distinguish between important and unimportant sentences and use that information to select

sentences for a summary. The classifier uses features based on word properties and sentence properties, such as the number of words in the sentence. Given the sentence importance classifier, two approaches to summarization are presented. In the first approach, the classifier is applied to the sentences in the input to determine their probability of being important. The sentences are ranked according to this probability, and the highest-ranking sentences are selected for a summary. In the second approach, they combine sentence importance with the idea that sentences at the beginning of a document are good candidates for a summary. Sentences are selected starting from the beginning of a document, but sentences are only included in the summary if they are classified as important. If a sentence is not important, it is skipped and the next sentence in the document is considered. These models outperform a baseline that selects random sentences as well as a state of the art summarizer. Incorporating sentence importance information in the model results in better performance than selecting sentences from the beginning of the document with no other information.

2.2 Graph-Based Summarization

In contrast to the previous methods that consider sentences individually, the following descriptions provide examples of how graph-based systems work by considering connections between sentences. In general, graph-based methods place emphasis on both coverage and redundancy by considering the similarity between sentences. One of the systems described in this section also focuses on improving coherence.

TextRank (Mihalcea and Tarau 2004) is a graph-based method for determining importance, which can be used for language processing tasks, including summarization.

Specifically, TextRank is used for single-document summarization. Graphs consist of vertices and edges that connect them. For summarization, vertices represent sentences. Sentences are connected in the graph when they are similar, and a relation between two sentences can be seen as a recommendation to refer to the other sentence for related information. Mihalcea and Tarau define similarity in terms of word overlap. They look at the number of words shared between two sentences, normalized for sentence length. The resulting similarity scores are used as edge weights in a weighted graph of the text. To determine which sentences to select for a summary, sentences are scored according to how many sentences they are connected to, the strength of those connections, given by the edge weights, as well as how connected the other sentences are. The sentences are sorted based on these scores, and the highest-ranking sentences are selected for the summary. When evaluated on the task of single document summarization, TextRank performed comparably to the top systems that participated in the Document Understanding Conference (DUC) 2002.

A related graph-based method, DivRank (Mei et al. 2010) uses not only prestige, or centrality in a graph, but also uses a preference for diverse coverage to inform sentence selection. This approach relates to the issues of coverage and redundancy discussed earlier. DivRank is motivated by the observation that simply choosing the most prestigious sentences that are highly connected in a network can result in choosing sentences that include redundant information and may not provide the most useful information coverage for a summary. The model is based on a time-variant random walk in the network, in which neighboring nodes compete with each other, with frequently visited nodes becoming more prestigious while their neighbors become less prestigious.

The result is that the nodes, or sentences, selected for a summary will not all come from one area of the network, producing summaries with more diverse coverage. Experiments showed that DivRank performs better than other graph-based methods on the task of summarization.

Christensen et al. (2013) also create a graph-based extractive summarizer, G-Flow, which performs extractive multi-document summarization. It combines the processes of sentence selection and sentence ordering in order to create more coherent summaries. G-Flow is an example of a graph-based system because it creates a multi-document discourse graph. Each vertex in the graph represents a sentence. An edge from one sentence to a second sentence indicates that the sentences have a discourse relationship. Specifically, an edge from s_i to s_j indicates that a coherent summary could contain s_i followed by s_j. Christensen et al. provide the following example.

s_1 Militants attacked a market in Jerusalem.
s_2 Arafat condemned the bombing.
s_3 The Wye River Accord was signed in Oct.

In this example, there would be an edge from s_1 to s_2, but no edges connecting these sentences to s_3. The system identifies pairs of sentences that have a relationship between them but does not need to label the exact relation because the graph only specifies whether there is or is not a relation. Several methods are used to determine the edges of the graph. One method takes advantage of the fact that events are often introduced with a verbal phrase ("attacked") and then referenced in later sentences with a deverbal noun ("the attack"). When pairs of sentences contain these verbs and nouns, edges are added to the graph between those sentences. In addition, a set of 36 discourse markers is used to

identify relations between sentences and add edges to the graph. These markers are used to find relations between adjacent sentences, as in the following example.

s_4 Arafat condemned the bombing.
s_5 **However,** Netanyahu suspended peace talks.

The authors also use a method to create edges between similar sentences, which is particularly relevant for multi-document summarization. Edges are constructed between sentences of different documents based on similarity between sentences, with similarity based on the presence of equivalent relations and arguments. Equivalence is determined using relational tuples according to Open Information Extraction (Banko et al. 2007). These tuples are of the form (argument1, relational phrase, argument2), such as (Militants, bombed, a marketplace). Sentences were judged as conveying similar information if their tuples contain synonymous verbs as well as at least one synonymous argument. This method detects that the following sentences contain similar information.

s_6 Militants bombed a marketplace in Jerusalem.
s_7 He alerted Arafat after assailants attacked the busy streets of Mahane Yahuda.

Another method for constructing edges is based on coreference. Edges are added to the graph when a sentence will not be understood without a previous sentence containing a referent such as the antecedent of a pronoun. Coreference is determined with Stanford's coreference system (Lee et al. 2011). Edges are weighted according to how many of these different methods predict a particular edge.

G-Flow attempts to overcome some of the problems associated with extractive summarization, specifically salience and coherence. Extractive summarization avoids the difficulties of generating grammatical natural language text, but extractive summaries often suffer from a lack of coherence between sentences. This problem results from how

most extractive systems work: sentences are individually selected from the original text and then combined into a summary without creating natural transitions between sentences. To overcome these challenges, using the constructed graphs, G-Flow constructs a summary by finding an ordered sequence of sentences that maximizes a joint objective function based on coherence, salience, and redundancy. Coherence is defined as the sum of the edge weights of adjacent sentences in the summary. Salience of the summary is the sum of the salience values of its sentences, where sentence salience is determined using a classifier that considers features such as sentence position, sentence length, and how often the nouns and verbs of the sentence appear in other sentences. Redundancy is determined using the relational tuples described above. The objective function finds a balance of these properties with the maximum summary length. The algorithm starts with the most salient sentence and then probabilistically adds sentences.

G-Flow was compared to state-of-the-art multi-document summarization systems, and the output was compared to human summaries and evaluated for aspects such as salience and coherence. Overall, summaries produced by G-Flow are preferred over summaries from other state-of-the-art systems. While it does not perform as well as human gold standard summaries, G-Flow is rated similarly to human summaries on several dimensions including coherence. Compared to another summarization system when a sentence reordering system was used, G-Flow summaries were still preferred, suggesting that a focus on coherence improves not only G-Flow's sentence ordering but also its sentence selection. One area that still needs improvement is overall information coverage. G-Flow's emphasis on qualities such as coherence and lack of redundancy results in less information coverage. Although there is room for improvement, G-Flow

goes beyond many other extractive summarization systems by focusing on creating more coherent and readable summaries.

2.3 Query-Focused Summarization

Many of the techniques used for generic summarization are also used for query-focused summarization. The main difference is that the sentences selected for the summary need to be biased toward a particular topic and should address the question that is asked. Given that goal, most methods approximate an answer to the query by finding sentences that contain words from the query or words that are related to the topic. For example, Vanderwende et al. (2007) use a query-focused system that increases the probability of words from the query in order to increase the likelihood of selecting sentences related to the topic.

Varadarajan and Hristidis (2006) also create a system for query-specific summarization. They find the parts of the text that are most relevant to the query and combine them. Similar to other systems, the document is represented as a graph. The text is divided into fragments, each node in the graph represents a fragment, and nodes are connected if the fragments are adjacent or are semantically related, as defined by containing the same words or synonyms from a thesaurus. Summaries are trees composed of a subset of the nodes and edges in the document graph. Summary scores are computed by considering the weights of the nodes and edges in the tree. Each node is scored based on its overlap with the query, and edges are weighted depending on the strength of the relationship. The words from the query are considered keywords, and the best summary is the best scoring tree from the graph that is total, meaning it contains all of the keywords from the query, and minimal, meaning that no nodes can be removed and still

have a total sub-tree. The use of keywords from the query differentiates this type of system from a generic summarization system. The keywords provide a way to select which information should be included in the summary, and in this case which nodes from the graph should be used to create the summary text. The quality of the summaries produced by this system was evaluated by human users who compared them to summaries created by other systems. The results suggest that people prefer the summaries generated by this system compared to other extractive systems.

2.4 Summarization Using Neural Networks

A lot of recent work on summarization has focused on techniques using neural networks. These methods have the advantage of requiring little task-specific knowledge because they learn from data, and they have been shown to perform well on many tasks. However, the models require large amounts of training data, and the models themselves can be difficult to understand and interpret (Li 2017). Despite these challenges, their strong performance on many natural language processing tasks has led to increasing amounts of research using these methods. This section describes how these models work in general terms and presents some of the summarization research that uses these methods.

Goldberg (2016) gives an overview of neural networks and how they are used for natural language processing tasks. He describes a neural network as a function $NN(x)$ that takes a d_{in} dimensional vector x as input and outputs a d_{out} dimensional vector. The input represents linguistic features including words and parts of speech. The features are chosen based on their relevance to the desired output. Each feature has a corresponding vector of values, the feature embeddings. The embeddings are typically initialized with

random values and trained as the network is trained. These feature vectors are combined to create an input vector, which is then given as input to the neural network. Commonly used features for neural networks are based on word embeddings (Bengio et al. 2003). Word embeddings are representations of words using vectors of real numbers. In contrast to sparse dimensional representations in which each word corresponds to its own dimension (with the total number of dimensions equal to the size of the vocabulary), neural networks use dense representations in which each word is a lower-dimensional vector, and similar words have similar vectors. For example, a feature corresponding to the word *dog* will have a similar vector to the feature for the word *cat*. The ability to have this shared information between features is an important benefit of these models. It allows for more generalization and the ability to find connections between similar features, such as relating *dog* and *cat* or treating singular nouns and plural nouns in a similar way.

The output of the network is also a vector. For example, for a classification task, the output would be a vector in which each dimension represents a different class. The values of the vector represent the strength of the association with that particular class, and these values can be used to determine the output class.

Between the input and output of the network are hidden layers. Each hidden layer is a vector of values based on the values of the previous layer and weight values, which represent the strength of the connections between the nodes in the two layers. Neural networks are also referred to as deep learning when the networks contain multiple hidden layers. The following diagram (Goldberg 2016) illustrates a neural network with two hidden layers. The nodes on the bottom represent the input features, and the nodes on top

represent the output features. Nodes in adjacent layers are connected by weighted edges. The values of the nodes are multiplied by the weights and run through a non-linear activation function to produce values for the next layer. The activation function allows the network to perform complex transformations of the input, other than linear transformations.

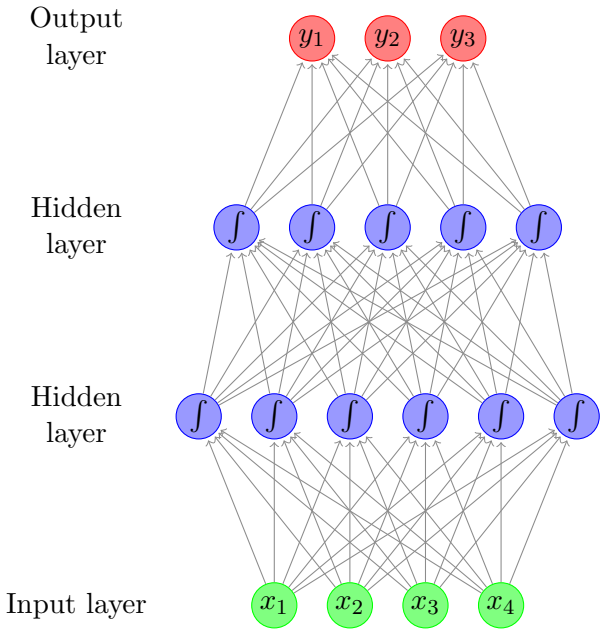


Figure 2.1: Neural network with two hidden layers (Goldberg 2016)

Neural networks are trained on supervised data by making repeated passes through the layers of the network and adjusting the weights to minimize the errors. A pass through the network produces an output that is compared to the correct output. The difference between the correct output and the predicted output is used to update the parameters of the network so that the output of the next pass through the network will be closer to the correct output. Once a network is trained, it can be used to predict the output of new data.

Neural networks have been used for different types of summarization, including single-document, multi-document, extractive, and abstractive. In addition to being used

as input to neural networks, word embeddings have also been used on their own for summarization, specifically for determining similarity. The rest of this section describes research that uses word embeddings and neural networks for summarization.

Zhang et al. (2015) use semantic information in the form of word vectors for summarization. They start with GloVe (Pennington et al. 2014) word embeddings as representations of word meaning, and they determine the meaning of a phrase, a sentence, or a document, by averaging the vectors of the words contained in that sequence. To create a summary, they find the averaged vector for the entire document as well as the vectors for each sentence in the document. The sentence vectors are compared to the document vector to determine the semantic distance between each sentence and the document. The sentences are ranked according to these distances, and the highest-ranking sentences that are most similar to the document are chosen for the summary. This method maximizes the semantic similarity between summaries and the documents they summarize. Zhang et al. find that sentences appearing in manual summaries have vectors that are more similar to the document vector than sentences that are not chosen for the summary, a finding that provides support for using this method.

Kageback et al. (2014) use phrase embeddings, to improve an extractive multi-document summarization system. They consider two methods for determining phrase embeddings. Like in the work just described, they combine the meanings of the words in a phrase by adding the word vectors. They also use an unfolding recursive auto-encoder, a more complex method that takes word order into account. Their results show that phrase embeddings improve summarization performance and suggest that this type of deeper semantic information is useful for the task.

In addition to using word vectors in the ways described above, some research focuses on the use of neural networks to classify whether sentences should be included in a summary (Cheng and Lapata 2016; Nallapati et al. 2017). For example, Cheng and Lapata (2016) use neural networks to perform sentence extraction for single-document summarization. Their sentence extraction process works by labeling sequential sentences as selected for the summary or not based on the features of the current sentence and the labels of the previous sentences, using a recurrent network. They also create a word extraction model, which is similar to language generation because instead of outputting a label for a sentence it outputs the next word to be included in the summary from the set of words contained in the document. The sentence extraction model performs better than or similarly to other state of the art systems without requiring linguistic annotation or manually created features. However, these models do require large amounts of training data in order to learn useful feature weights. The word extraction model does not perform as well, but it also faces a more challenging task than sentence labeling. For a fairer comparison, compared to a similar model in which words are generated from an open vocabulary, the proposed word extraction model that only generates words from the document achieves better results.

Other research focuses on going beyond extractive summarization. Several researchers use a neural network model to do abstractive summarization over single sentences by generating shorter headline versions of sentences (Rush et al. 2015; Chopra et al. 2016). Chopra et al. (2016) present a conditional recurrent neural network model that is trained to generate the next words for a summary based on the words in the input and the previously generated words. The model takes an input sentence and generates an

output sentence that is shorter but retains the meaning of the input. The following examples from Chopra et al. (2016) highlight some of the challenges of abstractive summarization.

> **Example 1:**
> Input: brazilian defender pepe is out for the rest of the season with a knee injury, his porto coach jesualdo ferreira said saturday.
> True Headline: football: pepe out for season
> Network Produced Summary: brazilian defender pepe out for rest of season with knee injury
>
> **Example 2:**
> Input: an international terror suspect who had been under a controversial loose form of house arrest is on the run, british home secretary john reid said tuesday.
> True Headline: international terror suspect slips net in britain
> Network Produced Summary: international terror suspect under house arrest

In example 1, the meaning of the input is preserved in the summary. However, it is not very much of a reduction from the input. It retains the entire first clause except for a few function words. Scaling this method for use on an entire document would likely result in relatively long summaries that are not much of a reduction from the original text. This highlights the challenge of finding a balance between conveying the information from the input and reducing the size of the input. The second example illustrates a different problem in which the meaning is not completely preserved. The focus of the input is that the suspect has escaped house arrest and is on the run, but the summary simply says that the suspect is under house arrest, leaving out the more important information. Comparing the actual headline to the produced summary also highlights the fact that actual headlines or summaries often contain different wordings from the input. In this case, the phrase "slips net" is a different way of phrasing the information in the input. However, the network generally produces summaries that contain the input text but with some words deleted rather than completely independent summaries generated from the meaning of the

text. Therefore, these approaches make progress away from extractive summarization, but true abstractive summarization remains a challenging task.

See et al. (2017) also perform abstractive summarization using neural networks. They focus on addressing several common shortcomings of these types of models, such as the problems seen in the examples above from Chopra et al. These shortcomings include producing details incorrectly such as leaving out negation when it was present in the original text or reversing the subject and object of a particular action. Another common problem is repetition when the model produces the same words or phrases multiple times. In contrast to some of the previously described research, See et al. create a model that takes news articles as input and produces summaries containing multiple sentences rather than taking one sentence as input and producing a headline. The core of their model is a pointer-generator network that has the ability to copy words from the input or generate new words from the vocabulary. The model learns a probability that determines whether the next word of the output should be copied or generated. This aspect of the model allows for words that appear in the input but are not part of the model's vocabulary to occur in the output, an important property for producing accurate summaries. Another important element of their model is that it takes into account what has already been covered in the summary. This is achieved by keeping track of how much attention the network has given the words in the input document. If words in the document have already informed previous choices of the model, they will receive less attention at later steps. This is important for reducing redundancy in the summary. Testing this model shows that it performs better than previous state of the art abstractive summarizers.

See et al. discuss the abstractive nature of their summarizer and some of the issues that relate to abstractive summarization in general. For example, there is a challenge of producing summaries that are faithful to the input but are still abstractions from the input. Their model allows for copying from the input, which increases its accuracy, but too much copying amounts to extractive summarization. The summaries produced by their model contain fewer novel n-grams and more copied sentences than the reference summaries they were compared to, indicating less abstraction. While their model does simply extract sentences in some cases, it also shows evidence of using several abstractive methods, including removing phrases and clauses and combining sentence fragments into new sentences. Overall, their model shows promising results for abstractive summarization but also highlights the challenges of true abstraction.

This section provides an overview of different methods for summarization. It describes different approaches to the task, including extractive and abstractive. It discusses a range of techniques, including simple heuristics like sentence position to neural networks trained on large amounts of data. It also demonstrates how different summarization systems tackle the challenges of coverage, redundancy, and coherence. The challenge of producing summaries that contain the optimal coverage of the input is motivation for the topic-based approach that was suggested at the beginning of this chapter and will be pursued in the rest of this dissertation.

The next section describes a way of conceptualizing summarization, specifically as a process similar to basic text compression, and discusses the connection between these two tasks.

3 Connection between Summarization and Compression

While most research on summarization assumes that a summarization system is given a target length or percentage of the original text's length when producing a summary, it is an interesting question to consider whether the ideal length for the summary of a given text could be automatically determined. Full-length texts often contain repetition. A certain level of redundancy is useful to support understanding of the main ideas of a text as well as to improve cohesion between different sections of a text. On the other hand, summaries are more constrained in terms of length and can also differ in purpose compared to full-length texts. The purpose of a summary is to provide an overview of the information in the original text and give a sense of what the original is about without including the same level of detail. Therefore, redundancy, while a useful strategy for creating understandable full-length texts, should be avoided when producing summaries. This leads to the interesting questions of how to determine redundancy and which information to leave out of a summary.

In addition to these questions about which content should be included in a summary, there is also the question of how long a summary should be. Most summarization systems specify the length with a number of sentences or a percentage of the length of the original text. However, another way to approach this issue is to say that a summary should be only as long as necessary to convey the main ideas of the text. This assumes that a text contains a core of crucial information and if that can be determined, a summary should contain only this information, with no superfluous information and no redundancy. The question of what the crucial information is in a text is not straightforward to answer. There could be different degrees of what counts as a main idea. If the goal is to produce a very short summary, such as a headline, what counts as

crucial information will be much more restricted compared to if the goal is to produce a more comprehensive summary. Therefore, while summary length could be determined by how many main ideas are in the text, it is also possible that as the allowed summary length changes, additional concepts in the text can be considered main ideas. It is an interesting question whether it is possible to determine some amount of crucial information in text and use that to inform summary creation.

This question relates to the issue of compression, in which information is encoded using fewer bits than the source. It is possible to perform compression without losing information by removing redundancy. This is considered lossless compression. There is also lossy compression, in which information that is determined to be less important is removed. Compression is used to reduce file size, which reduces the resources needed to use and store files. There is a lot of overlap between the task of compression and the task of summarization. Both tasks aim to produce a reduced version of the input that retains the important and non-redundant information. Because of this similarity, compression algorithms and the ideas behind compression can be used for summarization.

In addition to using the idea of removing redundancy when selecting content for a summary, another possibility is to use compression to determine how long a summary should be. A text that can be greatly reduced without loss of information should have a shorter summary than a text that cannot be reduced to the same degree. The length of a summary should be proportional to how much the original text can be compressed. A summary that is too long relative to its compression rate will likely contain redundant and unnecessary information. On the other hand, a summary that is too short relative to its compression may not contain enough information to reconstruct the main ideas of the

original text. For example, when considering two articles of the same length, it is possible that the ideal length for their summaries could greatly differ. Given all of the potential differences between texts including document type, author, and subject, among others, there is no reason to believe that a ten-sentence summary of one text is comparable in quality to a ten-sentence summary of a different text. It is worth considering that the length of a summary should depend on the original text. Therefore, when producing summaries, it may not be sufficient to simply stipulate the length of the summary. This is a different approach to the problem of summarization. Instead of treating it as a task of simply reducing a text to a specified size, this approach involves considering the ideal summary length for a given text, with this length depending on the content of the text itself. To my knowledge, there has not been much research focusing on automatically determining the appropriate summary length based on the original document, but it is an interesting question that relates to how summarization is connected to compression.

3.1 Related Research

Past research has explored ways that compression can be used for summarization as well as understanding how much redundancy exists in text and how much information is necessary to comprehend a document.

Grewal et al. (2003) investigate how compression software can be used for multi-document summarization. Specifically, they combine compression software with an automatic summarizer to see how much information is provided by different sentences. The summarizer used is MEAD (Radev et al. 2004), an extractive summarizer that uses features such as the ones described earlier in this chapter, including sentence length and position, overlap with a query, keyword matching, and centrality. They create a base

summary of a document using an extractive summarizer and then try individually adding each remaining sentence to the summary. They then compress these summaries and compare the lengths of the summaries and the sizes of the compressed files. Using this information, they can choose which sentence to add to a summary based on metrics such as which sentence contributes the largest increase to the size of the compressed file or which sentence results in the smallest ratio between the size of the compressed file and the length of the summary in characters. Among the metrics they considered, they found that the two metrics just described showed promising results and resulted in better summaries than summaries produced by the extractive summarizer alone. This illustrates one way that compression can be used for summarization, by using the sizes of compressed documents as a representation of how similar or distinct that document's sentences are. Even though file compression, such as the Lempel-Ziv algorithm (Ziv and Lempel 1978), typically operates at the character level and is therefore not necessarily expected to capture similarity of meaning, the results of this study suggest that this type of information does improve summary quality.

Witten et al. (1994) take a semantic approach to text compression with a form of lossy compression, meaning some information is lost and the original text cannot be completely reconstructed. Their method involves looking up words from a text in a thesaurus and replacing each word in the text with a shorter form of that word from the thesaurus. The intuition behind this method is that it shortens the text while maintaining the meaning. Here is an example of how this method works on a paragraph from their paper.

> **Original:** We have been struck by the apparent divergence between the research paradigms of text and image compression (Storer and Reif, 1991; Storer and

> Cohn, 1992), despite the fact that both are concerned with compressing information whose subjective quality must be recoverable. Schemes for text compression are invariably reversible or lossless, whereas although there certainly exist lossless methods of image compression, much research effort addresses irreversible or lossy techniques such as transform coding, vector quantization and fractal approximation.
>
> **Compressed:** We buy been struck by the due divergence mid the cut maps of bag and dud compression (Storer and Reif, 1991; Storer and Cohn, 1992), dig the root that both are dire and baring data whose biased bow must be recoverable. Lies for opus compression are invariably reversible or lossless, as as there fit be lossless uses of god compression, ton cut go heads irreversible or lossy ways will as go lawing, vector quantization, and fractal go.

While this technique does reduce the size of the text, the compressed version is much less readable and does not preserve the overall meaning. For example, the phrase "text and image compression" has been changed into "bag and dud compression." Another aspect of this approach to compression is that the number of words remains the same in the uncompressed and compressed versions. Depending on the goal of the compression, this aspect could be positive or negative. Summarization is a type of compression where the number of words is reduced. In contrast to this word-by-word compression that seeks to convey the same meaning as the original text by retaining all of the words in some form, summarization attempts to convey the same information as the original but with fewer words and sentences. Additionally, summaries are meant to be used by people. Simply reducing each word's length does not make a text shorter or easier to read for human users. In the example above, the compressed version may actually be more difficult to read because many words have been replaced with words that may be synonyms in a certain context but are not appropriate in this particular context. Therefore, this work on lossy semantic text compression suggests intuitions about how compression can be

performed in a way that takes semantics into account, but the specific approach needs to be improved to be useful for summarization.

Beyond compression at the character or word level, which creates a shorter representation of data from which the original data can be reconstructed, other research has considered the larger question of how to reduce texts through summarization but preserve the meaning of the original text. Along those lines, Morris et al. (1992) investigate how well summaries can be used in place of their source texts, specifically for the purpose of reading comprehension. They also consider the questions of how much information is necessary and where are the lines between insufficient information and information overload. Relatedly, there is also a question of when redundancy in text aids comprehension and when it unnecessarily increases processing time and effort. Morris et al. design experiments to see how well text extracts can be used to answer reading comprehension questions based on the original full-length texts. Based on suggestions from Shannon (1951) and Burton and Licklider (1955), they consider extracts that are 20% and 30% of the original text. Shannon (1951) and Burton and Licklider (1955) explored the redundancy of text by testing people's ability to predict the next character in a text given the preceding characters in order to estimate bounds on the entropy and redundancy of text. They estimated that text has a redundancy of around 75%, meaning text can be reduced by 25% without loss of information. However, much of this reduction is likely due to removing letters from words or eliminating predictable words rather than capturing more complex measures of redundancy in meaning. However, given that suggestion, Morris et al. chose 20% and 30% as extract sizes because of their proximity to the theoretical limit of 25%, which represents how much a text could theoretically be

reduced without loss of information. If there is a difference in performance between the two sizes of extracts, specifically lower performance on extracts containing 20% of the original, that result would suggest that there is support for this limit.

Extracts were constructed by determining the important sentences using word and sentence features such as those discussed above. When comparing the mean differences between these different conditions, there was no significant difference in performance between the full text, the 30% extract, and the 20% extract. These results have several interesting implications. They suggest that summaries in the form of extracts can successfully be used instead of their source texts without decreasing comprehension. The results also show that 25% is not a strict reduction limit. The 20% extract was as useful as the 30% extract as well as the full text. Morris et al. point to a suggestion by Kibby (1980) as an explanation for this finding. Kibby suggests that the proposed 75% redundancy rate may indicate the amount of redundancy at the levels of words and characters but may not apply to larger units of meaning. This difference between the lower levels of characters and words and the higher, more abstract level of meaning is the core issue in connecting summarization to compression.

Simovici et al. (2015) explore how compression can be used to judge similarity between texts. They propose a method for using compression in document classification. They use the Lempel-Ziv-Welch lossless file compression algorithm (Welch 1984). They define a measure of document dissimilarity as shown in the following equation, where x and y are documents.

$$d(x, y) = \frac{C(xy)}{C(x)+C(y)} \quad (2.6)$$

$C(x)$ is the size of the compressed document. The closer the value of this measure is to 1, the more dissimilar two documents are. In that case, the compressed size of the two concatenated documents is the same as the sum of the compressed sizes of the individual documents, meaning no additional reduction in size takes place when the documents are combined, indicating that they are not very similar. On the other hand, the value of this measure for two similar documents is closer to 0.5. For example, if the documents are identical the size of their compressed concatenation should be the same as their individual compressions, leading to $\frac{size}{2size}$, a value of 0.5. Simovici et al. propose that this measure of dissimilarity can be used to help cluster documents and also to evaluate the quality of summaries or abstracts of documents. The intuition is that the dissimilarity between abstracts should be correlated with the dissimilarity between full documents. For summarization it is very useful to have a way to determine similarity between parts of a text in order to group related sentences and to reduce redundancy between sentences selected for a summary. This dissimilarity measure can serve as a starting point for using compression for summarization. In chapter 4, this measure will be discussed in the context of topics and how topics can be used for summarization. Specifically, it will be used for comparing different sentence groupings, or topics, to see which groupings are the most distinct, based on the idea that divisions of sentences into groups that are more dissimilar to each other can better capture the full range of topics discussed in a document. Using distinct topics should be useful for producing summaries that convey the same information as the original text and finding a balance between information coverage and redundancy.

There are obvious similarities between compression and summarization, as discussed above. It is worth exploring how and whether both the intuitions behind compression and the details of how compression is implemented can be useful for performing summarization. One major difference between compression and summarization is the level at which they are typically performed. Most compression algorithms operate at the string level, finding characters and sequences of characters that are repeated within a text. Given the goals of compression, operating at the string level is very effective. The goal of compression, lossless compression in particular, which is typically used for text compression, is to reduce the size of the input data in a manner that allows the original data to be reconstructed. At the string level, in a text of a reasonable size there will generally be repetition because the text is composed of a finite number of characters. In contrast to the surface string level, which requires no special processing, summarization can be thought of as operating at the meaning level. In theory, summarization involves compressing or reducing the ideas within a text. The goal of summarization is to convey basically the same information as the original text or at least the most important or relevant information. However, determining when two sections of text are redundant in terms of their meaning is a more difficult task than simply finding repeated characters. In addition, the distinction between lossless and lossy compression is particularly relevant for summarization. There is a question of whether it is possible for an original text to be perfectly reconstructed from a summary or if a summary is necessarily a form of lossy compression.

Operating at the meaning or idea level requires going from the string level to a representation of the string's meaning and performing compression over the meaning.

Just as texts contain repeated occurrences of the same characters, they also contain repetition at the meaning level. This repetition allows texts to be cohesive and be "about" something. While character repetition can easily be found by humans or computers, repetition at the meaning level is harder to determine. Use of the same words can be an obvious indication of repetition, but there are also different words that convey the same meaning and words that are related because they are used to describe concepts in the same general domain. Some sentences have no overlapping words but are related because they appear in the same paragraph or section. Given the variation in how the same information can be conveyed, it is a challenging question to determine whether different pieces of a text are similar or redundant. If there were a simple answer, summarization could be completely reduced to a compression problem.

3.2 Explorations of Information Found by Compression

An important step in understanding how the ideas and algorithms behind compression can be used for summarization is to understand exactly what information from a text is captured by compression. In order to do this, I used a compression algorithm to compress texts and then look for correlations between compression ratios and different text features. These features include the total number of words, characters, and bytes in a text as well as percentages comparing the number of unique words, stopwords, nouns, and verbs to the total number of words. These features were chosen as low-level properties of a text that could be considered an approximation of the amount of distinct content contained in a text and could therefore potentially influence how much a text could be compressed.

The compression method I used is gzip, a compression algorithm based on the LZ77 algorithm (Ziv and Lempel 1977). The basic idea of the algorithm is to find repeated strings in the input. When a string is found to be a repetition of a string found previously in the data, the second instance of the string is replaced with a pointer to the first occurrence. This pointer indicates the distance to the previous instance as well as the length of the repeated string. A few short examples[1] show how this algorithm works. A short text such as "abcdabcd" contains 8 total characters, but the second 4 characters are identical to the first 4 characters. Therefore, this text can be reduced to a compressed representation, "abcd[D=4, L=4]". This pointer indicates that the repetition begins by going back 4 characters (distance = 4) and that the number of characters to be repeated is 4 (length = 4). Another text that shows how this algorithm works is "Blah blah blah blah blah!", which is very compressible because it contains one word repeated several times. Taking one character at a time, the first six characters (including the space) are all distinct, non-repeated characters, "Blah b". Looking at the next few characters, the repetition begins. Not only has the next character, "l", been seen previously but the entire sequence of "lah b" is repeated. The string "Blah blah b" can therefore be compressed and the second occurrence of "lah b" represented with a pointer. The representation of this sequence is "Blah b[D=5, L=5]", where D=5 indicates that the repetition begins 5 characters before the end of the string (at "l") and L=5 indicates that the number of characters to repeat is 5 ("lah b"). Continuing with the rest of the string, the next characters are also repetitions. This example shows a useful aspect of the algorithm, which is that the length given in the pointer can be greater than the distance. In that case,

[1] http://www.gzip.org/deflate.html

characters are repeated until the specified length is reached. The entire string of "Blah blah blah blah blah!" can be represented with "Blah b[D=5, L=18]!". This method is how many compression algorithms work.

The dataset used to explore compression is the texts from the RST Discourse Treebank (Carlson et al. 2002), specifically the training set, which consists of 347 Wall Street Journal articles. The texts range in length from 26 words to 1916 words, with an average length of 455 words.

The first comparison, shown in the following plot, involves looking at the number of words a text contains and that text's compressibility. Comparing the compression ratio to the total number of words, at first there is a steep increase in compression ratio as the number of words increases, and then the compression ratio begins to flatten out and remain in the same range even as the number of words increases. This suggests that there is a limit to how much a text can be compressed. An extremely short text cannot be compressed very much because it cannot contain as much repetition as a longer text. This is seen on the plot below where the texts with the lowest word counts have a low compression ratio. As texts get longer, the amount of compression rapidly increases up to a point. The compression ratio levels out between 50% and 60%, with texts of length 500-2000 words having compression ratios in this range. The second plot shows the log of the word count compared to the compression ratio. As the plot shows, this relationship is roughly linear, confirming that differences in word count affect the compression ratio more for shorter documents. As documents get longer, the raw differences in word counts create smaller differences in compression ratio.

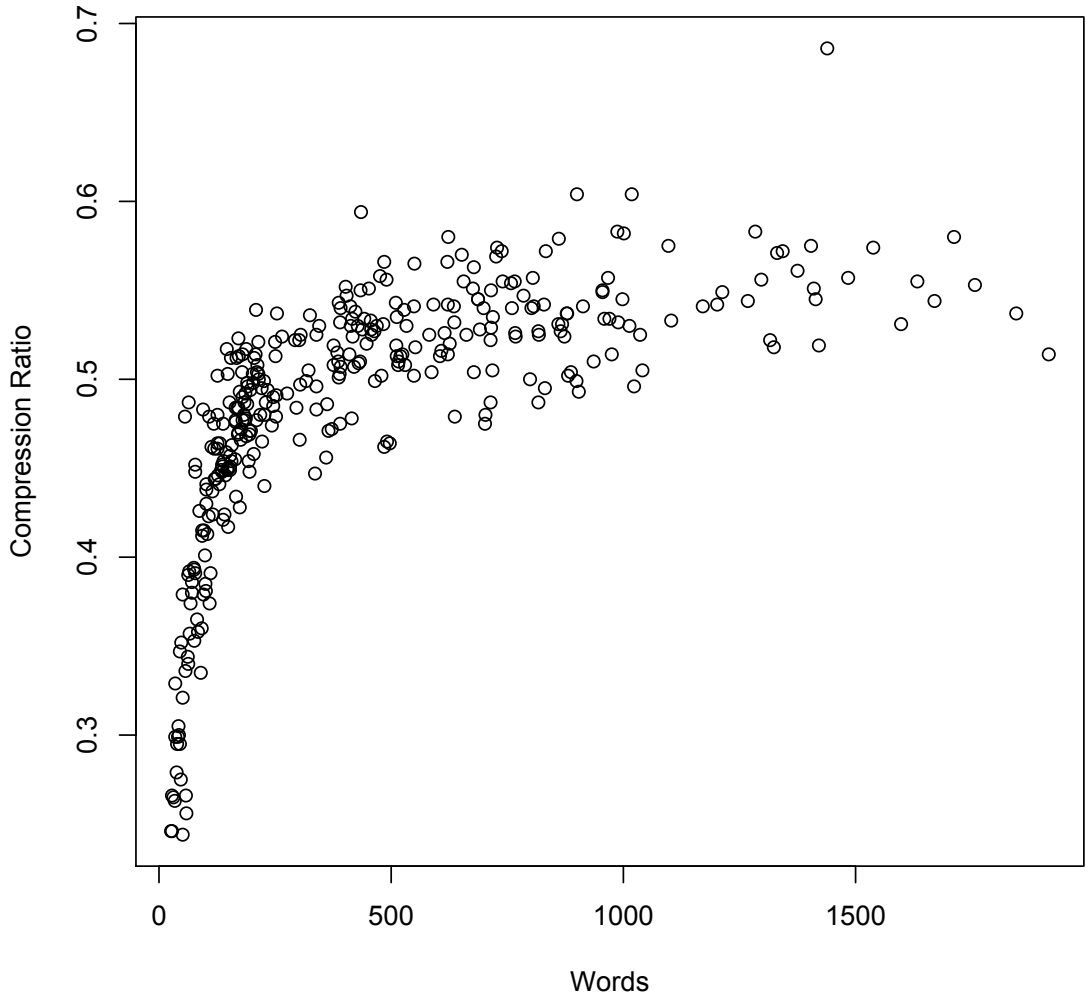


Figure 2.2: Word count compared to compression ratio

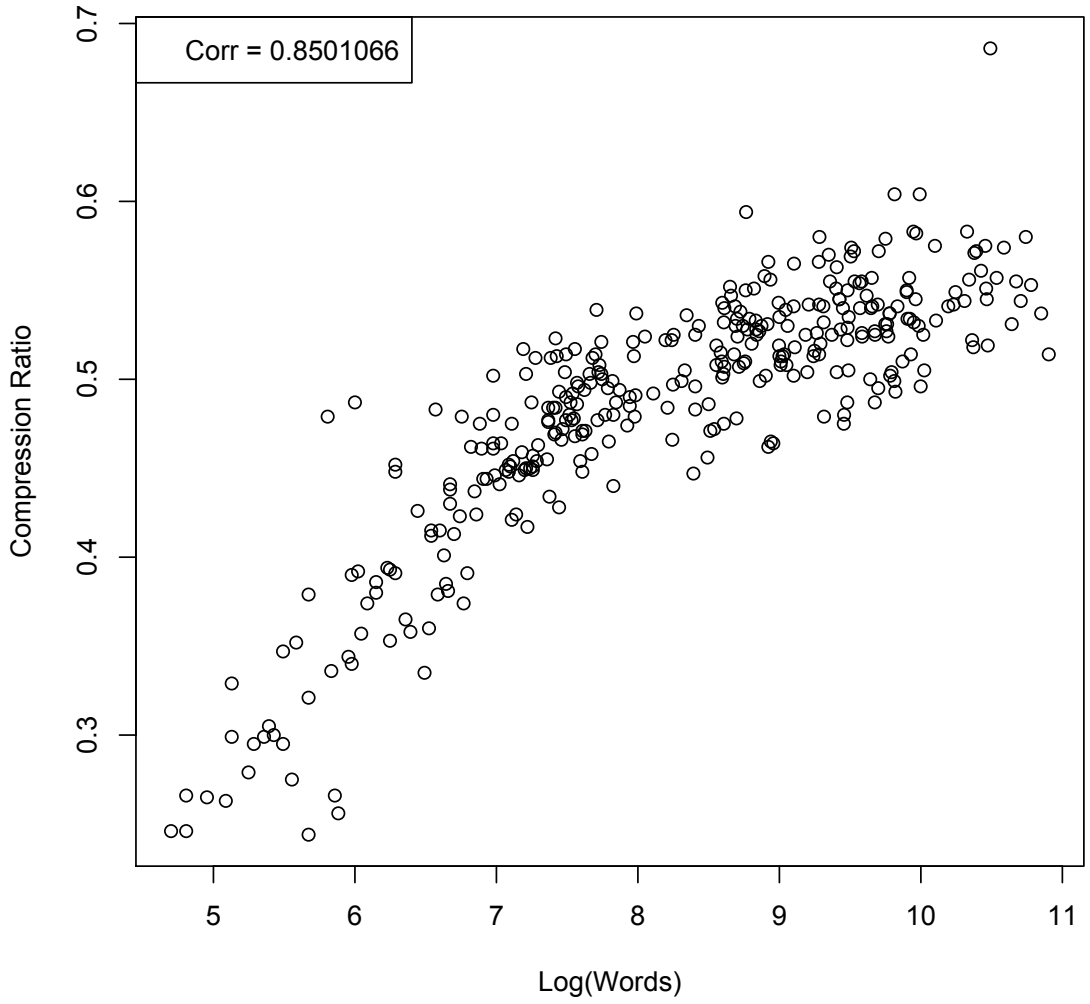


Figure 2.3: Log of word count compared to compression ratio

While the last plot considered all words in a text, the following plot explores the relationship between compressibility and unique words. Unique words are words that appear in the text only one time. This plot shows that the amount that a text can be compressed is almost entirely explained by how many unique words it contains. If a text has a small number of unique words relative to its total word count, it is more

compressible as shown on the left side of the plot. Texts that contain a high percentage of unique words with few word repetitions cannot be compressed very much, as shown by the points in the bottom right corner of the plot.

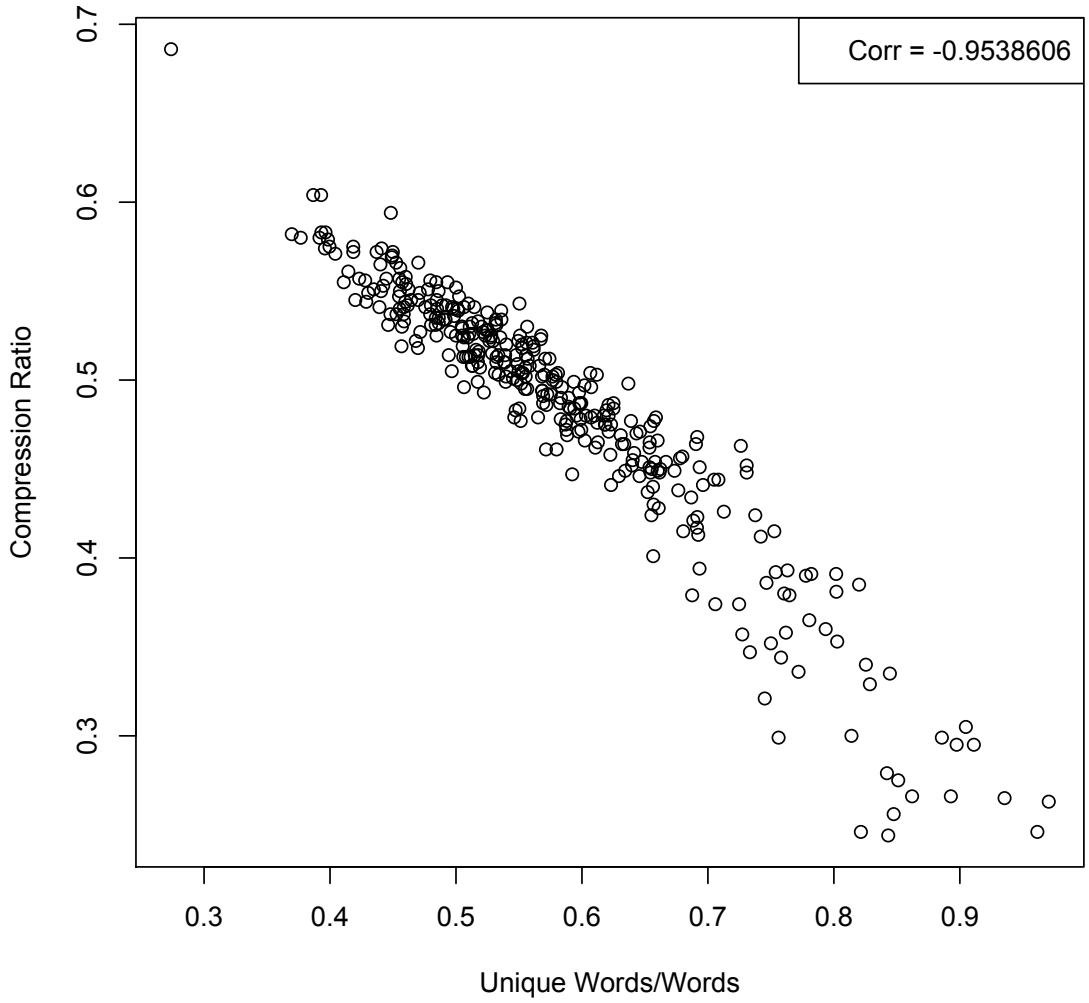


Figure 2.4: Percentage of unique words compared to compression ratio

The following plot shows the log of the word count compared to the percentage of unique words. It is similar to the plot showing the log of word count and compression ratio, which were positively correlated. In this case, the log of word count is negatively

correlated with the percentage of unique words. As documents get longer, they tend to contain fewer unique words.

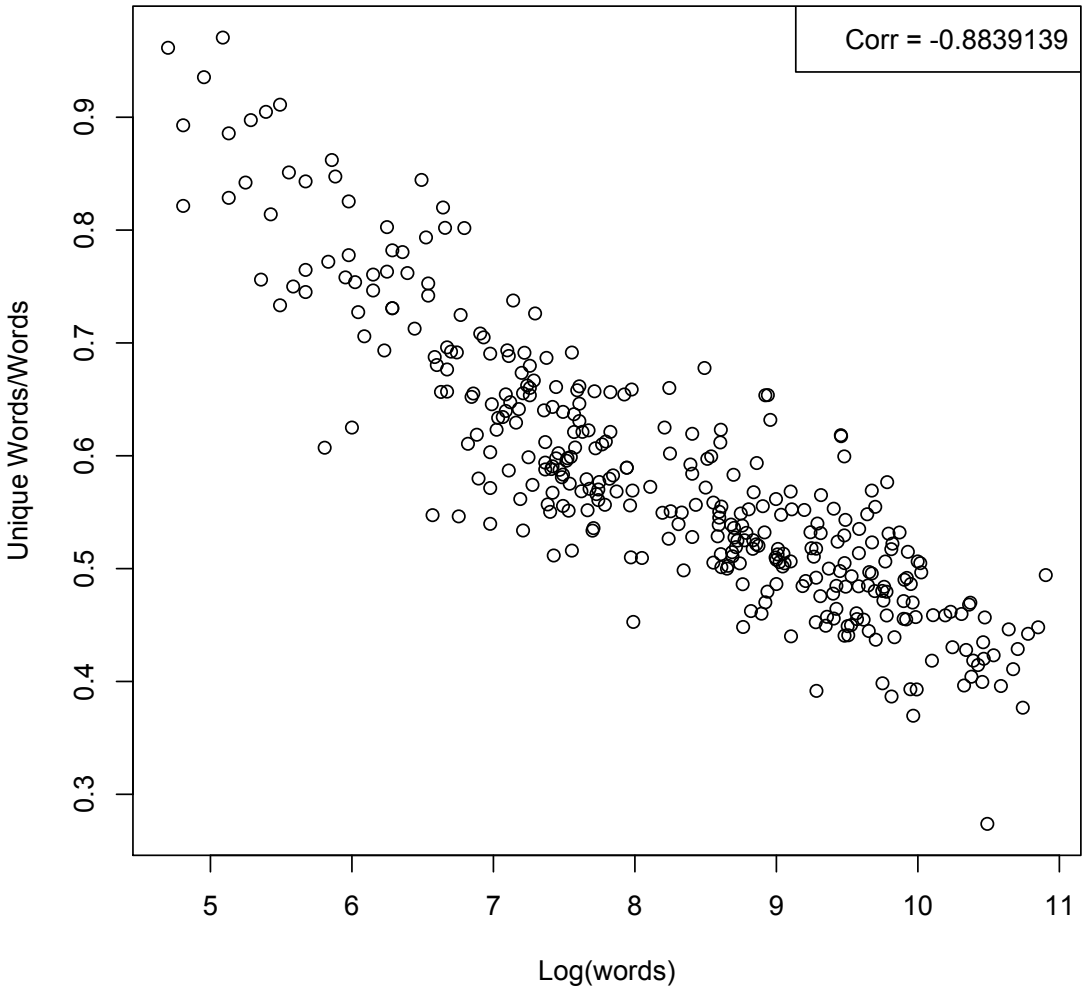


Figure 2.5: Log of word count compared to percentage of unique words

Given the results described above, compression algorithms clearly capture information at the word level, but there is a question of whether this type of compression captures any higher-level information. If the compressibility of a text correlates with aspects of a text related to its meaning, then compressibility can be used instead of a more complex representation of meaning for summarization. One text feature to consider is word order. Word order is important for certain aspects of meaning, so it is useful to know whether word order affects how much a text can be compressed. In order to investigate this, I created scrambled versions of the texts in the RST corpus. The words in each text were randomly scrambled into a different order, so that the meaning of the text is no longer preserved. The following is an example of a short text in its original version and scrambled version.

> **Original:** Richard W. Lock, retired vice president and treasurer of Owens-Illinois Inc., was named a director of this transportation industry supplier, increasing its board to six members.
>
> **Scrambled:** increasing of its Inc., this president a board members. Lock, supplier, retired vice treasurer Owens-Illinois named Richard six director W. and was to industry of transportation

The scrambled version of the text contains the exact same words as the original version, but by changing the word order the text has lost the meaning of the original version, providing a test to see whether this change in meaning has any effect on compressibility. Both versions of the text were compressed. The correlation between the compression ratios for each version are shown in the plot below.

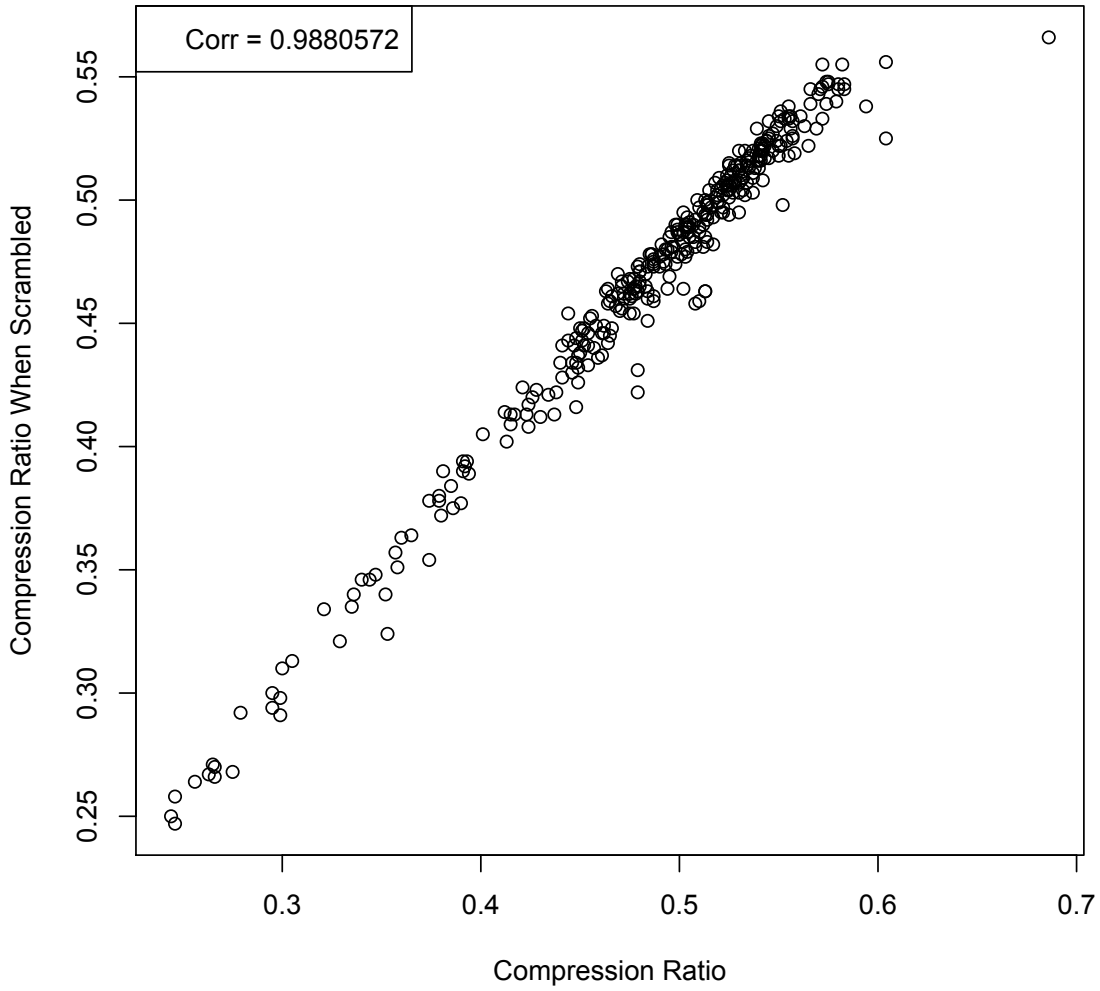


Figure 2.6: Compression ratio compared to compression ratio when text is scrambled

As the plot shows, the compression ratios are very strongly correlated. These results suggest that changing word order has no effect on compressibility, and the correlations between texts and their compression ratios seem to be limited to the word level. It is possible that a compression algorithm could capture patterns in text at higher levels such as phrases or sentences, but this does not appear to be the case since the

results are basically identical regardless of word order. If there were repetition of longer units such as phrases, there should be a benefit from maintaining consistent word order. The lack of this improvement suggests that the type of redundancy found in texts does not arise from simple repetition of strings. For summarization, information about text meaning is crucial. Therefore, the basic string-based compression algorithms used here are not satisfactory for determining how much distinct content a text contains. In order to make that determination or some approximation of it, a representation of meaning is required.

Overall, this analysis demonstrates several interesting findings about compression. There is a very strong correlation between the amount a text can be compressed and the percentage of unique words it contains. This suggests that the patterns and redundancy found by typical compression algorithms can be reduced to the word level. Effective summarization requires a measure of information content and redundancy at the meaning level. Due to the relative simplicity of string-based compression, it would be useful to be able to use the compressibility of a text as a heuristic for how much distinct content that text contains and therefore how much information a summary should contain. The analyses described in this section were performed with that motivation in mind. Most of the results point to the inadequacy of typical compression algorithms to capture the type of information that would be useful for summarization. Therefore, this work confirms an interesting correlation between compression and the word level and provides motivation to extend the ideas and intuitions of compression beyond the word level in order to use these ideas for summarization.

At the beginning of this section, the issue of summary length was discussed, as it relates to compression. Determining the ideal length for a summary is not an easy task. In general, summary length should be related to the content of the document being summarized, with documents containing many distinct concepts requiring longer summaries than documents with fewer concepts. Compression provides a way to conceptualize the problem of reducing a text to its most important information. However, extending the ideas of compression to summarization requires going beyond finding similarities at the character level, as typical compression methods do, and using a more abstract representation of meaning. The appropriate representation of meaning as well as methods to determine similarity in meaning are challenging questions that require further research. An intermediate approach between using string-level compression and meaning-level compression can be used. For example, one of the main ideas of compression is to remove redundancy so that what is left is the set of distinct information in the text that cannot be reduced any further. Extending this idea to summarization means that summaries should include coverage of all distinct concepts in the text while limiting redundant coverage. The next chapters expand on this idea by describing the idea of dividing texts into topics that can be used for summarization to ensure that summaries contain the important information from the original documents.

4 Conclusion

This chapter describes previous techniques used for summarization. It introduces extractive summarization, which is the type of summarization used in the current work. Approaches ranging from fairly simple methods such as word frequency or sentence position to more complex methods such as neural networks are discussed. This chapter

provides an overview of the research in this area. In addition to the general discussion of previous methods, the connection between summarization and compression is explored. The similarity between these two tasks provides motivation for using the intuitions and principles of compression for summarization. The idea of using topics for summarization was introduced as a way to implement some of the ideas of compression, specifically reducing a text to its distinct, non-redundant information. The next chapters will build on this suggestion of topics and provide an in-depth exploration of how texts are composed of topics and how that information can be used to improve the summaries produced by an extractive single-document summarization system.

Chapter 3

Notions of Topic

1 Topic Definitions

1.1 Overview

When using topics for summarization, one important consideration is how topics are defined. There are different notions of what it means to be a topic. For example, given a document with labeled sections, each section could be considered a topic. In that case, a scientific article could contain topics including background, methods, results, and discussion. Another definition of topic is more semantically-based, with texts divided into topics based on the similarity of the words they contain. This chapter explores previous definitions of topic, the types of methods used to automatically determine topics, and the possibilities of using different notions of topic for automatic summarization.

The rest of this section introduces definitions of topics from the literature. Section 2 discusses motivation for using topics to structure text based on studies of how people read and understand text. Section 3 describes previous approaches for taking the abstract notion of topic and implementing a method to automatically determine the topics in a text. Section 4 describes Rhetorical Structure Theory, the basis for the topic division method proposed in this work. Explorations of how to use RST for summarization,

including methods from previous research, are discussed. Section 5 concludes the chapter.

1.2 Definitions

In linguistics, there are different notions of what it means to be a topic (Gundel and Fretheim 2004; Lambrecht 1996; Gundel 1988; Blei 2012; Griffiths et al. 2005; Griffiths et al. 2007; Guo and Diab 2012; Van Dijk 1977; Van Kuppevelt 1995; Asher 2004; Chafe 1994). Topics have been defined as the "thematic continuity of discourse segments that span more than a single sentence" (Asher 2004) as well as an "aggregate of coherently related events, states, and referents that are held together in some form in the speaker's semiactive consciousness" (Chafe 1994). Notions of topic differ on whether they operate at the sentence level or the discourse level and whether they are based on structural or semantic information.

A lot of research on topics (Gundel 1988; Gundel and Fretheim 2004; Lambrecht 1996) focuses on sentence-level topics, specifically topic-comment structure. In that structure, a topic is defined as an entity that a sentence increases knowledge about or requests information about. A comment is a statement made relative to the topic. In simple terms, a topic is what a sentence is about, and a comment is a statement or question about that topic. Specific constructions can be used to refer to topics in the discourse, and in English these involve moving the topic to the front of the sentence. This is seen in the following examples in which the topics (in bold) appear at the beginning of the sentence.

That store, it's still open.

Because she didn't have the ingredients, she went to the store.

This definition of topic captures the idea that topics are what a piece of text is about. However, this definition operates at the sentence level, and therefore does not capture the types of topics that are relevant for describing an entire text. The types of topics that should be useful for summarization are topics at the level of the entire text that describe not only what a single sentence is about but rather what groups of sentences and larger sections of text are about. Sentence-level topics could be extended to describe larger texts by looking at whether the sentence-level topic changes or remains the same between sentences. Topics could be described as sequences of sentences that maintain the same topic.

Other work has suggested ways to conceptualize topics beyond the sentence level. Van Dijk (1977) suggests that sentence topics linearly connect pieces of information, such as a topic and comment, while higher-level text topics provide a global organization of the text. They ensure a sense of coherence between different parts of a text by providing an overall proposition that sentences within the text are related to. This difference between sentence topics and text topics affects how topics are described. While the topic of a sentence might simply be a noun phrase such as *Eva* or *Prague*, the topic of an entire text, in terms of what that text is about, is more likely to be a proposition, such as *Eva went to Prague*. These text topics capture what is most important in a text and answer the question of what a text is about at the macro-level. Expressing the topics in words in effect produces a summary of the topics. This highlights the connection between topics and summarization and motivates why topics are useful for this task. Topics are a way of capturing the important ideas that persist throughout a text.

Van Kuppevelt (1995) also extends the topic-comment structure beyond the sentence level, suggesting that discourses consist of a large hierarchical topic-comment structure. Specifically, discourse is organized around topics. The following definition of topic is given:

> "The notion presupposes that a discourse unit U - a sentence or a larger part of a discourse- has the property of being, in some sense, directed at a selected set of discourse entities (a set of persons, objects, places, times, reasons, consequences, actions, events or some other set), and not diffusely at all discourse entities that are introduced or implied by U. This selected set of entities in focus of attention is what U is about and is called the topic of U."

This definition agrees with an intuitive description of a topic. A text or discourse contains several different ideas, and parts of a text relate to different ideas from this set. A topic includes the parts of a text related to a particular idea. In Van Kuppevelt's approach, determining topics involves identifying questions in the discourse. The topic of a discourse unit depends on the question, explicit or implicit, that the unit answers. A discourse consists of questions, which introduce topics and comments that answer those questions, and a topic is closed when a question has been answered. Topics are parts of a discourse that are related through the entities and ideas they discuss, and as these ideas are sufficiently discussed and new entities introduced, the topic changes.

Generalizing from this idea, topics often create boundaries between sections of text based on whether the following text builds on what has previously been discussed or introduces a new question into the discourse. These topics shifts can be signaled by using particular sentence constructions or discourse markers. The topic may be assumed to stay the same in other situations such as when pronouns and referring expressions are used because these expressions rely on antecedents and previous information for interpretation. In this case, topics are defined by the presence of novel information.

This section has suggested several different ideas of what it means to be a topic. In this work, the definition of topic used is the one described above for discourse topics or topics of texts. The topics of interest characterize larger sections of text beyond the sentence level. Following the definitions given above, a topic is what a section of text is about and the ideas that are in focus in that section. Given this overall definition of topic, there are several ways to understand and determine the topics in a text. There is an important distinction between the abstract notion of topic and the methods used to determine those topics from an input text. Two main notions of topic are considered in this paper, and both sides of this issue will be addressed. Both notions of topic agree with the above definition but use different characteristics of text to find and describe topics in text. One is based on relations from Rhetorical Structure Theory (RST), and the other is based on word distributions through Latent Semantic Analysis (LSA). The difference between these two notions of topic is largely an issue of granularity as well as the methods used to determine the topics. RST topics operate narrowly by capturing topic changes within a single text, and topics consist of contiguous sets of sentences. In terms of determining these topics, texts are organized and annotated with relations according to RST, and these labeled relations inform the topic divisions. On the other hand, LSA topics can be applied on a broader scale to find the topic of an entire document or the topics present in a collection of documents. On the practical side of determining topics, LSA uses word distributions to find the topics present in a document or collection of documents and relate pieces of text to those topics, with no constraints on topics containing adjacent sentences. These two notions of topic represent different ways of making use of the information in text to divide documents into sections that are about the

same concept. Both of these notions of topic and the information they use to determine the topic structure of a text will be discussed in detail in later sections of this chapter.

2 Motivation for Topics from Human Processing Studies

Different theories of text structure and organization have been proposed. An interesting question to consider is how people structure text when writing as well as how they create a structure of a text when reading and processing it for different tasks such as summarization. Some research has focused on understanding how people process texts and which aspects of a text affect how people create a mental structure of the information in texts.

Lorch et al. (1985) conduct experiments to see how reading times are affected by the topic structure of a text. The topics they consider include properties of different countries such as geography, climate, and politics. Topics correspond to paragraphs in a text and are introduced by topic sentences, such as "The geography of Morinthia is particularly rugged." They found that people read sentences introducing a new topic faster when the topic was closely related to the previous topic. They also found that reading times were faster when people were presented with an outline of the text's topic structure before reading it. These results show that people are sensitive to topic structure and use this information for reading and comprehending text. This suggests that from a processing standpoint, topics are a useful way for readers to organize texts.

Johnston and Afflerbach (1985) study the processes people use to determine which information in a text is important and what the main ideas of a text are. Specifically, they asked people to read a text that did not contain any topic sentences and make a statement about the main idea of the text. This information was collected through

verbal reports from the participants, who were instructed to think aloud and describe the processes they were using while reading and determining the main idea. Based on these reports, Johnston and Afflerbach found that people use several types of clues to determine importance including their prior knowledge of a topic as well as specific cues based on the words and structure of the text. These strategies include noticing repeated mentions of the same or similar concepts. Another strategy was to use words that explicitly signal relations between parts of a text, such as words like "similarity" or "later." In terms of more structural cues, people commented on using the relationships between paragraphs and knowledge of how particular types of texts, such as scientific articles, are structured to determine which information was important. People also showed sensitivity to what they believed were the intentions of the author of a text, suggesting that it is important to consider the goals and intentions of a writer when choosing what to include in a text as well as how to present it. This is related to theories of discourse including Rhetorical Structure Theory (Mann and Thompson 1988), which will be discussed in more detail below in terms of how it can be used for automatic summarization.

Hyönä et al. (2002) used eye-tracking experiments to see where people focused their attention when reading a text, including looking at which portions of a text were re-read and at what point they were re-read. They divided participants into groups based on the results of the eye-tracking experiment and found that one group of people was particularly sensitive to the topic structure of a text, with topic structure indicated by several types of sentences including headings, first sentences of sections, and last sentences of sections. For the group that was more sensitive to topic structure, more of

their fixation time was spent looking at section headings in the text as well as looking back at headings. This group also focused more attention on sentences at the end of a topic than other participants. As a later part of the experiment, all of the participants were asked to write a summary of a text, and judges evaluated these summaries for which topics were included in them. The summaries were scored based on the percentage of topics included and how well the order of presentation of the topics correlated with the order of appearance in the original text. Based on these measures, the participants who were sensitive to topic structure when reading a text produced the best summaries by including more topics and more accurately capturing the order of topics. Lorch et al. (1987) found similar results when looking at the outlines produced by people who were sensitive to topic structure. Those people who paid more attention to this structure tended to produce better outlines than people who did not. Hyönä et al. conclude that the topic processing strategy was the most successful strategy for reading a text and then summarizing it. Indeed, it is an interesting result that people use different processing strategies and this impacts how they choose content for a summary. However, they only evaluate summary quality based on how well a summary captures the topic structure of the text. Therefore, a reading strategy that relies on processing topics would be expected to perform better.

This experiment suggests an interesting relationship between where attention is focused during reading and which information should be included in a summary. Focusing on information related to topic structure when reading a text results in producing better coverage of topics when writing a summary. As discussed in previous chapters, a good general-purpose summary should include broad coverage of topics, with

an emphasis placed on covering more topics in less detail rather than covering fewer topics in more detail, which agrees with how the experimenters in this study evaluated summary quality. Therefore, according to this measure, the use of topic structure for summarization is motivated based on the summaries produced by this strategy as well as the fact that it is a strategy used by people when reading and summarizing text.

If topic structure aids in processing and comprehension, then using topics for summarization is motivated. Although not performed by people, automatic summarization can be thought of as similar to the task of humans reading a text. A person reading a text must understand how different pieces of a text relate to each other in order to determine the overall meaning of the text. To do this, they use strategies like the ones discussed here, such as noticing topic changes and creating a structural representation of the text. The task of a summarization system is similar. It must take an input text and determine the information that should be contained in a summary. In this way, it needs to "understand" the overall meaning in order to create a useful summary that best conveys the information in the original text. Therefore, the same strategies that have been shown to influence human reading and processing in experimental work can be used in summarization systems. Considering the task from this side as well as the more theoretical side should help in creating a summarization system that performs similarly to people. The goal of a summarization system is to create summaries that are useful to people, so a system should create summaries that agree with human ideas of importance and text structure. Given this understanding of how people use topics, the next section considers methods for dividing texts into topics automatically.

3 Previous Approaches for Determining Topics

3.1 Topic Segmentation

While there are many definitions of what it means to be a topic, there are several general approaches to finding topics and dividing texts into smaller units that correspond to topics. These methods are used when topics are needed for tasks such as information retrieval and summarization. One way of approaching this problem is topic segmentation. Similar to other types of segmentation, the task of topic segmentation is to determine where to place boundaries within a continuous stream of text. Instead of the boundaries representing breaks between words or sentences, the boundaries correspond to shifts in topic. The goal of topic segmentation is to produce a linear partition of the text. Each section produced by segmenting a text corresponds to a topic, and a text is composed of a sequence of topics. This method assumes that topics are contiguous sections of text. Topic segmentation seeks to find the boundaries between these sections and create a topic structure for the text.

A lot of work in this area focuses on a distributional notion of topic by looking at the similarity of words before and after potential boundaries to determine where a shift in topic is likely to occur. Hearst (1997) proposes a method called TextTiling that divides texts into passages or subtopics using word co-occurrence patterns. The model works at the paragraph level by dividing paragraphs into topics. The model is based on the idea that a change in topic should be accompanied by a change in vocabulary, and therefore looking at how word frequencies and distributions change within a text should provide an indication of where topic changes occur. Words that are very frequent or uniformly distributed are unlikely to provide much information about topic structure, so the task is

to find less frequent words that occur in groups or clumps and use the boundaries between these groups as boundaries between topics. Given this concept of what a topic shift should look like, the main method Hearst considers for how to perform segmentation involves comparing blocks of text for the similarity of their words. Boundaries between sentences are given scores, and text blocks with many words in common are given higher scores. Low scores between adjacent blocks indicate that a topic shift is likely. Specifically, valleys where two blocks have a low score and are surrounded by higher scores are potential boundaries and are scored based on the depth of the valley. Depending on the desired number of boundaries, boundaries are placed at positions with the highest depth scores. To evaluate this method of topic segmentation, the output of the system was compared to human judgments of which paragraph boundaries involved a topic change. The TextTiling model performed better than a baseline that randomly places boundaries between paragraphs, but it does not perform as well as human judges. This work explores the intuition that topic changes are accompanied by changes in the vocabulary being used, and therefore that topics are associated with particular words. The results show support for this hypothesis and show that word similarities and distributions are useful for creating a segmentation of a text.

Choi (2004) describes a linear topic segmentation model using sentence similarity. Specifically, sentences are compared to each other using cosine similarity, and these values are used to create a similarity matrix. This matrix is then modified based on the intuition that short pieces of text do not contain enough information for the value of a similarity measure to be a reliable indicator of importance. Therefore, Choi proposes using the value of a similarity measure as an estimate of similarity, for example by

indicating that sentence 1 is more similar to sentence 2 than it is to sentence 3. The values in the similarity matrix are replaced by a rank representing a sentence's local importance in context. The rank is determined by counting the number of neighboring cells in the matrix containing a lower similarity value. Given these rank values, segmentation is performed by a process of divisive clustering where all sentences begin in one cluster, which is divided into smaller clusters to maximize within-cluster similarity until a threshold is reached. Testing this method showed improved performance compared to other segmentation algorithms and showed the effectiveness of using ranked similarity values rather than absolute values when working with small text segments.

Du et al. (2015) explore the use of ordering information for topic segmentation. Their work is based on the tendency for documents to include topics in a particular order to increase comprehension, with certain orders occurring frequently in documents from the same domain. For example, Wikipedia articles about major cities tend to include certain topics such as geography and history, and many articles also present these topics in the same order. Du et al. use a generative model that takes a set of topics and a set of documents and determines both which topics to discuss in a document as well as how those topics should be ordered. They test their model on datasets that contain documents from the same domain with patterns for topic ordering as well as datasets that do not have any regularities in how topics are ordered. On datasets that include ordering regularities, this segmentation method performs better than models incorporating no ordering information. However, this method does not perform as well when the documents follow no ordering patterns. These results demonstrate that when available regularities related to topic ordering can be used to improve topic segmentation models and more generally the

results show the effectiveness of using a document's structure and domain-specific knowledge to inform tasks such as topic segmentation.

While text segmentation is generally linear, some work has focused on creating a hierarchical topic structure (Yaari 1997; Eisenstein 2009). Yaari (1997) takes the same approach as Hearst (1997) using lexical cohesion between paragraphs as a measure of coherence. Paragraphs are then grouped into larger structures using a hierarchical clustering method that successively groups together the most similar segments. Yaari proposes that the hierarchy created by this clustering represents the internal structure of a text. Yaari finds that the text boundaries determined by this hierarchical method align better with human judgments of boundaries than the linear methods while also providing a richer structure of the text than a linear segmentation. Eisenstein (2009) used the same lexical methods as linear segmentation to create a hierarchical segmentation. His work is based on the idea that texts are cohesive at different levels of granularity with a small set of words indicating a low-level subtopic and a larger set of words indicating a higher-level topic that spans multiple subtopics. In his model, a word can be generated from the word distribution of any topic that it is contained within, a low-level topic or a higher-level topic. The hierarchical algorithm was tested on its ability to determine chapter and section boundaries in a textbook and found that lexical features could successfully be used to perform hierarchical segmentation in addition to linear segmentation. Eisenstein points out the issue of how this model could be extended to text segments of different granularities, specifically smaller segments such as paragraphs. This is an important question, particularly in the context of summarization, where the texts to be summarized may vary in length. For summarizing texts such as academic articles, news articles, or

encyclopedia entries, the text segments of interest are generally paragraphs or sentences. For text segments of this size, there may not be enough lexical information to learn from and use to accurately determine topic boundaries. This motivates thinking beyond comparisons at the word level for topic segmentation, specifically for shorter texts where lexical information may be too sparse to be the only source of information.

3.2 Topic Modeling

In addition to work on topic segmentation, a lot of research focuses on a notion of topic based on a distributional model of word meaning. The idea is that the meaning of a word can be explained by the contexts in which that word occurs. Words that occur in similar contexts are likely to have similar meanings. The meaning of a word can be thought of as a combination of the meanings of the documents in which it occurs, and the meaning of a document can be thought of as a combination of the meanings of its words. This is the underlying idea of topic modeling methods, which are used to find similarities between texts and group texts into topics.

As an example of how these methods work, one type of topic modeling technique is Latent Semantic Analysis (LSA) (Deerwester et al. 1990; Landauer et al. 1998). LSA represents words as points in a high-dimensional space, represented with a matrix, and the size of this space is reduced from a matrix in which each word has its own dimension in the matrix to a smaller set of dimensions so that each dimension in the matrix is not a word but rather a concept that could relate to many words. In this way, documents can be thought of not only in terms of the specific words they contain, but in terms of higher-level concepts that those words relate to. This method addresses the fact that two documents may be similar in terms of content and meaning even if they do not contain

the same words. Judging document similarity only at the level of words ignores synonymy, the fact that different words can have the same or similar meanings. LSA goes beyond the word level and finds the concepts within documents. These concepts are the topics that a document is about, and therefore LSA provides a way to conceptualize what topics are, and it can be used as a method for finding the topics of a text.

In more detail, LSA is a mathematical method that takes text as input and infers vector representations of words that are intended to correspond to their meaning. A corpus of texts is represented as a matrix in which each row is a unique word and each column is a segment of a text – a sentence, a paragraph, a document. A cell in the matrix contains a frequency value for how often the word in that row occurs in the text segment of that column. The next step of LSA is to apply singular value decomposition (SVD) to the matrix. This involves deconstructing the matrix into the product of three other matrices. The three matrices produced by SVD are U, S, and V', where U and V are the left and right singular vectors and S is the diagonal matrix of singular values. When these matrices are multiplied, the original matrix is produced. The dimensionality reduction step of LSA involves keeping only some of the values of S and removing the others. When the matrix is reduced in this way, multiplying the three matrices will result in an approximation of the original matrix. If the S matrix represents concepts, only the k largest values of this matrix are kept, where k is the number of dimensions or concepts that the data should be reduced to. The results are modified versions of the three matrices: $t \times k$, $k \times k$, and $k \times d$. Thinking about this model in terms of its use on text, the $t \times k$ matrix is a term by concept matrix, $k \times k$ is a concept by concept matrix, and $k \times d$ is a concept by document matrix. When these matrices are multiplied, the result is

a new $t \times d$, term by document, matrix that is an approximation of the original matrix. The values of the matrix are no longer actual frequency counts but are instead estimated frequencies. The changes in these values should reflect latent similarities between terms. A term that did not occur in a particular document may now have a higher frequency estimate for occurring in that document if that document contains other words that are related in the semantic space. Similarly, a word's frequency in a document may be reduced in the estimated matrix if given the other words in the document that word is unexpected and uncharacteristic of the rest of the document. Therefore, this method takes actual frequency counts of words in documents and uses word similarity to estimate how frequently we would expect to see a word in a particular document given the concepts in that document. Landauer et al. (1998) provide a way to conceptualize this process: "This text segment is best described as having so much of abstract concept one and so much of abstract concept two, and this word has so much of concept one and so much of concept two, and combining those two pieces of information (by vector arithmetic), my best guess is that word X actually appeared 0.6 times in context Y." At this point, documents can be examined to see their distribution of topics, and documents can be compared to each other to see how similar they are in terms of the topics they contain.

LSA uses mathematical techniques to create a semantic notion of topic. It uses only the information given in the input text and requires no outside resources or information to determine topics. Based on word frequency, it ignores word order, syntax, and other notions of sentence or document structure.

Given this specific explanation of how one of these topic modeling methods works, it is worth considering what types of information these models capture at a more

general level as well as the types of tasks they are used for. These methods define topics in distributional terms. Topics can be identified by a clustering of related words and determining the topic of a section of text involves comparing the distribution of words in the text to the clusters of words that define the topics. The topics found by these models depend on the set of texts given as input to the model. They find topics that characterize the entire collection. The total set of topics may therefore not be relevant for every individual document, but each document can be thought of as having a different proportion of each topic with some topics more represented in the document than others. Topic modeling methods are used to create a semantic space and locate texts within that space. Labeling a text with a topic or dividing a set of texts by topic depends on additional steps that are performed using the information from this semantic space.

The following section describes how the abstract information given by these methods is used for different tasks. Some of the most common uses are to find documents most related to a query or classify documents into different categories. Deerwester et al. (1990) use LSA to determine how relevant different documents are to a query. This is accomplished by comparing a query vector to document vectors and ranking documents according to their similarity to the query. On this task, LSA has been shown to perform better than basic term matching for information retrieval. Zelikovitz and Hirsh (2001) use LSA for text classification. They find how similar unlabeled texts are to labeled examples in each class and use that information to determine the class of the unlabeled texts. Foltz (1996) uses LSA to assess text coherence. Sentence vectors in the reduced semantic space were compared between adjacent sentences in a document and averaged to get a coherence value for the document.

Riedl and Biemann (2012) build on the ideas of topic segmentation with TextTiling (Hearst 1997) by incorporating information from the topic modeling method of Latent Dirichlet Allocation (LDA) into their model called TopicTiling. LDA is a generative model in which topics are considered to be distributions over a fixed vocabulary, and different documents contain topics in different proportions. Riedl and Biemann use the same idea of comparing adjacent pieces of text to determine their similarity. Specifically, they calculate a coherence score between adjacent blocks of text. Instead of basing this score on occurrences of the words themselves as in TextTiling, Riedl and Biemann use LDA to reduce the sparse word vectors into smaller topic vectors, where words are replaced by the topic they were assigned by LDA. Blocks of text are then represented by a vector the length of the number of topics in the LDA model rather than a vector the length of the number of words in the vocabulary. The coherence between blocks of text is calculated as the cosine similarity of these smaller topic vectors. Local minima in the coherence scores are considered possible segmentation boundaries. After testing this model, they report improved performance over other topic segmentation models, including TextTiling. This result shows that information from topic models can be used to improve topic segmentation.

All of these tasks benefit from using information beyond the surface word level. This is a key strength of topic modeling methods. By moving from the word level to the meaning level, connections are found between texts that appear different on the surface and texts can be characterized by the concepts they contain.

Some research has focused on combining topic information such as that from topic modeling methods with rhetorical information (Chen et al. 2016; Ó Séaghdha and

Teufel 2014). Chen et al. (2016) combine topic information and rhetorical structure into a single model, recognizing the importance of both of these types of knowledge for understanding the structure of a document. Specifically, their goal is to model a document's intent structure by assuming that documents contain two types of words: topic words and rhetorical words. They build on two areas of research and ideas of text structure. The first area is topic modeling in which documents are assumed to contain a mixture of topics. The second area is rhetorical structure such as that described by Mann and Thompson (1988) in which sentences have a rhetorical function or intent. This type of structure will be described in detail in the next section. Chen et al. assume that topics are relatively stable within a document while the rhetorical functions of sentences change and tend to follow a certain order throughout the course of a document. They propose a model that considers both topic and rhetorical structure. The words of a document are assumed to be either topic words related to the subject matter of the document or intent words that signal the rhetorical function of a sentence, such as to indicate a cause or a result. They consider intents to be at the level of the sentence and follow a certain order within the document, while topics are at the document level. They use a generative process that generates documents from a topic distribution and an intent distribution. For each word, they determine whether it is a topic word or an intent word and either draw a word from the intent distribution or draw a topic from the topic distribution and a word from that topic. Their model performs better than several baselines on the task of labeling sentences with intent labels. The combined model performed better than a model that considered only the intent structure. This result shows the effectiveness of using both topic and rhetorical information in combination.

4 Rhetorical Structure Theory as a Basis for Topics

4.1 Rhetorical Structure Theory

In contrast to the distribution-based notion of topic described in the previous section, the notion of topic pursued in this work is based on Rhetorical Structure Theory. Rhetorical Structure Theory (RST) is a framework for describing the organization of a text and what a text conveys by identifying hierarchical structures in text. Such structures are intended to represent the organizational principles that characterize documents. Pieces of text relate to each other in different ways in order to accomplish the writer's purpose with some pieces more central than others. Beginning with the clause level, relations can hold between successively larger spans of text forming a hierarchical structure of how all spans of the text are related to each other.

Mann and Thompson (1988) describe how texts are analyzed according to this framework. One intuition behind RST is that the text structure itself conveys information beyond the information explicitly asserted by clauses in the text. Specifically, Mann and Thompson say that text structure conveys "relational propositions." These propositions are not expressed in the form of clauses like other propositions but concern the relationship among clauses or groups of clauses. According to Mann and Thompson (1988), "Relational propositions, therefore, challenge theories of language that equate the communication effect of a text with the 'meanings' of its sentences and compose those meanings from the meanings of its syntactic structures and lexical items" (260-261).

Relations connect two non-overlapping pieces of text, and their combination conveys information beyond that of the individual clauses, such as the relational proposition that the information from one clause is evidence for the information in the

other clause. Relation types include enablement, circumstance, background, justification, and evidence, among others. An important part of RST relations is the distinction between nuclei and satellites. The nucleus of a relation is one of the spans of text connected by the relation that is more essential to the purpose of the writer and is comprehensible on its own without the satellite. The satellite is the element that generally cannot appear on its own but provides some type of supporting information for the nucleus. The effect is that the reader has an increased belief in the information given in the nucleus. As an example of what these relations are intended to convey, an evidence relation connects one span of text to another span that is intended to increase the reader's belief in the first span. In this example of an evidence relation from Mann and Thompson, the information in the satellite about the program producing the correct calculation provides evidence for the nucleus which states that the program works.

> **Nucleus:** The program as published for calendar year 1980 really works.
>
> **Satellite:** In only a few minutes, I entered all the figures from my 1980 tax return and got a result which agreed with my hand calculations to the penny.

The evidence relation and relations in general capture information about how different pieces of the text connect to each other and work together to achieve the writer's purpose. Using relations, all spans of text are related to each other. RST follows the constraints of completeness, connectedness, uniqueness, and adjacency, meaning all units are connected as minimal units or constituents, a unit participates in only one relation, and no units overlap with each other. The following example illustrates how text units are connected through relations to form a discourse tree.

Example of RST relations (from Thompson and Mann 1987)

1. There is a gardening revolution going on.

2. People are planting flower baskets with living plants,
3. mixing many types in one container for a full summer of floral beauty.
4. To create your own "Victorian" bouquet of flowers,
5. choose varying shapes, sizes, and forms, besides a variety of complementary colors.
6. Plants that grow tall should be surrounded by smaller ones and filled out with others that tumble over the side of a hanging basket.
7. Leaf textures and colors will also be important.
8. There is the silver-white foliage of dusty miller, the feathery threads of the lotus vine floating down from above, the deep greens, or chartreuse, and even the widely varied foliage colors of the coleus.

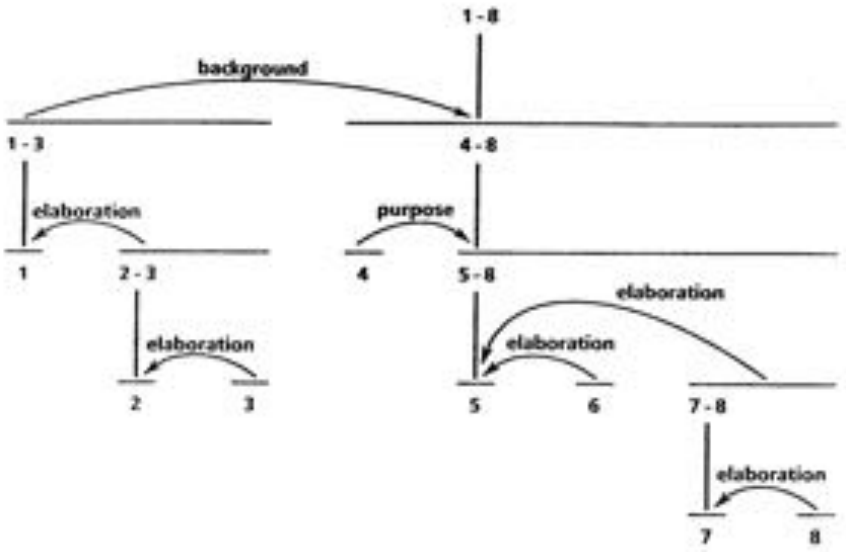


Figure 3.1: RST structure

This example shows how a hierarchical structure is created using RST relations. Elementary Discourse Units (EDUs) are the building blocks of RST structure. They are the lowest level units that are arguments of RST relations. EDUs are typically clauses.

The EDUs combine and these larger units are combined with other clauses or discourse units, until all units are connected in a single tree. In this example, clauses 2 and 3 are in an elaboration relation with clause 3 providing additional information about the topic of clause 2. Similarly, clauses 7 and 8 are also in an elaboration relation. These relations are at the lowest level, between clauses. In addition to clause-level relations, RST involves connecting larger spans of text. For example, the discourse unit consisting of clauses 2 and 3 serves as the satellite of an elaboration relation with clause 1. Larger and larger spans are created and connected through relations until all units are connected. In this case, the highest-level relation is a background relation that connects the unit comprising clauses 1-3 to the one comprising clauses 4-8. RST results in structures such as this one in which all units in the text can be connected into a tree.

Mann and Thompson (1988) distinguish between two types of relations: subject matter and presentational. Subject matter relations connect spans of text in terms of how their content is related. For example, cause relations indicate that the subject matter of one span is the cause of something in the subject matter of the other span. On the other hand, presentational relations refer to the presentation of the information and serve to increase an inclination in the reader, such as increasing belief or acceptance. For example, evidence relations involve the use of one text span to increase belief in another span. Other research has explored this distinction in relation types. Moore and Pollack (1992) refer to these types as informational and intentional and suggest that these are distinct levels of analysis, with an understanding of discourse relying on both levels. Moser and Moore (1996) also refer to the informational and intentional structures of discourse. Both of these levels of information are important parts of understanding how

texts are connected, and RST makes use of both of these relation types to characterize the connections between sections of a text.

RST is a way to understand how parts of a text are connected to each other. All units in a text are connected in a hierarchical structure so that sections of text of all sizes participate in relations. The types of relations in RST capture how units of text are connected in terms of the purpose and organization of the text.

4.2 Exploration of Effects of RST Structure on Summary Creation

Given that RST provides useful information about how texts are composed and how pieces of text of different sizes, down to the clause level, are connected, I explored how this information could be used to inform a summarization system in its choice of sentences to include in a summary. The first focus was looking for connections between the sentences that are included in gold standard summaries and the RST properties of those sentences. In order to investigate which parts of a text are likely to be selected for a summary, a classifier was trained on labeled data to differentiate between text units that were selected or not selected for a summary. The data came from the RST Discourse Treebank (Carlson et al. 2002), which contains Wall Street Journal articles annotated with RST structure. In the RST Treebank, there are 150 documents that have manually-created extractive summaries. To create the summaries, each document was summarized by two people, producing two summaries for each document, creating a total of 300 extractive summaries. These summaries were created by extracting Elementary Discourse Units (EDUs). The summarizers were instructed to select a number of EDUs from the document based on the square root of the total number of EDUs in that document.

In order to classify units as selected or not selected for a summary, the NLTK Naïve Bayes classifier (Bird et al. 2009) was used. I created a set of features based on RST information. The features are described below.

The following features capture characteristics of the EDUs.

Nucleus/satellite: This feature is whether a unit is a nucleus or a satellite of the relation it participates in. The distinction between nuclei and satellites is an important aspect of RST annotation, and is likely to be relevant for summarization because it is meant to capture importance, with nuclei more important than satellites.

Relation type: Each relation between two units in a text is labeled with a relation type that describes how the two units are connected. Relation types include elaboration, consequence, comparison, and background relations, among others. These relation labels provide useful information because they describe the type of information contained in the text and how the different parts of a text interact. This feature captures the relation type of the relation of which the unit under consideration is the nucleus or satellite.

First word of relation type: Several relation types have sub-classes. For example, within the set of "elaboration" relations, there are "elaboration-additional" and "elaboration-object-attribute" relations. Looking at the first word of the relation type allows for generalization across larger classes of relations. As with the previous feature, this feature captures information about the relation that the relevant text unit participates in.

Word count: This feature considers the number of words in a text unit, and is meant to find any patterns in whether longer or shorter units tend to be chosen for summaries.

Contains a cue word: There are several words that signal the presence of a specific relation or introduce new or important information. I took the following list of cues from the RST Discourse Treebank manual: after, although, as, because, by, despite, following, however, if, meanwhile, since, so, until, when, while, without. This feature looks at whether a text unit contains one of these cue words to see whether these words affect whether a unit is selected for a summary.

Distance from root: This feature looks at the depth of a unit in the tree, in terms of how far it is from the root. This is an attempt to find whether being higher or lower in the tree affects a unit being selected or not.

Mononuclear or multinuclear: Most relations have one nucleus and one satellite, but in certain cases when nodes are equally important, a relation may have more than one nucleus. This feature considers whether a text unit is part of a mononuclear or a multinuclear relation to see whether the number of nuclei impacts whether a unit is selected.

The following features consider how text units are related to other units.

Parent nucleus/satellite: Each text unit has a parent node within the RST tree structure. This feature looks at whether a text unit's parent is a nucleus or satellite. This captures another aspect of how important a unit is within the larger tree structure.

Parent relation type: This feature captures the relation type of the relation in which a unit's parent is the nucleus or satellite. This feature is based on the idea that children of certain relation types may be similar to each other.

Parent span size: Internal nodes in the tree have different span sizes, meaning the number of nodes that appear below that node, as children or children of children. At the lowest level, a node may have two leaf children, and therefore only two nodes below it. At the highest level, the root node has all other nodes in the tree below it. Each leaf node is labeled with a number, and each internal node is labeled with a span, such as 7-8 or 18-22. I calculate span size as the difference between the two elements of the span. For these examples the span sizes would be 1 and 4. This feature provides some information about how high in the tree a unit is, as well as how large the other argument of its relation is. If unit 1's parent has a span size of 16, that means the other element of the relation is 2-17, which is a larger portion of the tree. Therefore, unit 1 may be an important unit because it is related to a large section of the tree.

Whether the other argument is a leaf or interior node: This feature looks at the other argument of the relation that the text unit under consideration participates in, and considers whether it is also a leaf node or whether it is an interior node.

The classifier was tested using these features. The training data consisted of 120 WSJ articles from the RST corpus with 2 summaries for each article, totaling 240 summaries. For each summary it is known which units were selected and which ones were not. The task of the classifier is to learn how to predict whether a unit will be

selected or not based on the RST features described above. The classifier was tested on a set of 15 documents, with a total of 30 summaries. Performance is measured with precision, recall, f-score, and accuracy. Each of these measures captures a different aspect of how the classifier performs. Accuracy is a simple measure of how many examples were classified correctly.

$$(3.1) \quad Accuracy = \frac{true\ positives + true\ negatives}{true\ positives + true\ negatives + false\ positives + false\ negatives}$$

The problem with using accuracy alone is that it is very dependent on the balance between true positives and true negatives in the data. If the correct label for most of the examples is positive, then a classifier will achieve good accuracy by simply predicting positive for every example. Precision and recall take more information into account and can be more useful for evaluating performance. Precision measures how many of the classifier's positive predictions were in fact true positives.

$$(3.2) \quad Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

On the other hand, recall measures how many of the true positive examples the classifier correctly identified.

$$(3.3) \quad Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

F-score combines precision and recall into a single measure. Looking at all of the measures in combination provides a more complete picture of how a classifier performs. The results of the classifier using several different feature combinations are presented in the following table.

N/S	Rel	Rel-F	WC	Cue	Dist	Mo/Mu	P N/S	P Rel	Span	L/I	Prec	Rec	F	Acc
x	x	x	x	x	x	x	x	x	x	x	0.5	0.187	0.272	0.892
	x	x	x	x	x	x	x	x	x	x	0.517	0.183	0.271	0.893
x		x	x	x	x	x	x	x	x	x	0.443	0.119	0.188	0.888
x	x		x	x	x	x	x	x	x	x	0.52	0.159	0.244	0.893
x	x	x	x	x	x	x		x	x	x	0.492	0.187	0.271	0.891
x	x	x	x	x	x	x	x		x	x	0.528	0.174	0.262	0.894
x	x	x	x	x	x	x	x	x		x	0.385	0.113	0.175	0.885
x	x	x		x	x	x	x	x	x	x	0.527	0.180	0.269	0.894
x	x	x	x		x	x	x	x	x	x	0.5	0.187	0.272	0.892
x	x	x	x	x		x	x	x	x	x	0.359	0.086	0.138	0.885
x	x	x	x	x	x	x	x	x	x		0.508	0.183	0.270	0.892
x	x	x	x	x	x		x	x	x	x	0.508	0.187	0.273	0.892
x	x	x			x	x	x	x	x	x	0.527	0.180	0.269	0.894
x	x	x	x	x	x	x					0.390	0.098	0.156	0.886
x	x	x						x			0.296	0.049	0.084	0.885
	x	x						x			0.321	0.055	0.094	0.885
x	x	x									0.269	0.043	0.074	0.884
x		x									0.167	0.003	0.006	0.890
	x										0.5	0.003	0.006	0.892

Table 3.1: Results of Naïve Bayes classifier with different RST features; feature abbreviations are defined below

N/S: nucleus or satellite Rel: relation type Rel-F: first word of relation type WC: word count

Cue: contains cue word Dist: distance from root Mo/Mu: mono- or multi-nuclear P N/S: parent nucleus or satellite

P Rel: parent relation type Span: parent span size L/I: other argument leaf or interior

The first line in the table shows that using all of the features results in precision of 50%, recall of 18.7%, f-score of 27%, and accuracy of 89%. Accuracy is very high, but as discussed above, this measure is somewhat misleading. There are many more true negatives than true positives in the data, so predicting negative for every example would result in high accuracy. In fact, in this case, predicting negative for every example would also result in accuracy of 89%. Therefore, while accuracy is very high, it is actually no higher than simply classifying every example as part of the same class. Looking at the other measures, precision is 50%. About half the time when the model predicts a unit should be selected for a summary, it is correct. Recall is fairly low at 19%, meaning there are many selected units that it does not correctly identify. For comparison, if we used the model that predicts negative in every case, recall would be 0 and precision would be undefined. This shows the problem of looking at accuracy alone and shows that the classifier does better than a very simple baseline. However, the performance is still fairly low and this classifier could not be used on its own to reliably predict whether a unit should be included in a summary.

The rest of the table shows performance when different subsets of the features are used. Lines 2-12 show the results when a single one of the features is removed. The remaining lines show various other combinations, such as only features corresponding to relation type or only features corresponding to the unit itself without considering its parent or sister argument. The highest precision, 52.8%, is achieved when the feature for parent relation type is not included. Similar performance is achieved when the features for word count and the presence of cue words are not included. The best recall achieved by the classifier, 18.7%, is much lower than the best precision. The best recall is found

when all features are included in the model as well as when several features are removed. Removing features either lowers recall or keeps it the same. Recall never increases. Interestingly, when the only feature used is relation type, the model achieves 50% precision, which is the same as when all the features are included. However, that model achieves the worst recall, only 0.3%. Therefore, relation type on its own is not sufficient for classifying units as selected or not selected. No matter which features are used by the classifier, accuracy never drops below 88%.

In addition to the Naïve Bayes classifier, a Maximum Entropy classifier was also tested. The results from that model when all of the features were used were a precision of 75%, recall of 8%, f-score of 15%, and accuracy of 90%. This model had problems similar to those of the Naïve Bayes classifier. It was able to achieve a fairly high precision but at the expense of recall. The model did not select many units and instead classifies almost all units as not selected. The resulting f-score was lower than many of the scores reported above. The rest of the explorations described below used the Naïve Bayes classifier.

In order to better understand whether information from RST is useful for deciding which parts of a text should be included in a summary, I created a classifier with more basic features that do not make use of the structure given by RST. The features used were word count, whether a unit is the first unit of the text, whether a unit is in the first ten units of the text, and the average word frequency of the words in the unit. This classifier had a precision of 32.6%, recall of 13.1%, f-score of 18.7%, and accuracy of 87.7%. These scores are all lower than the scores achieved by the classifier with RST features, suggesting that RST information is more useful for this task than other basic text features.

I also tried combining these text features with the RST features to see what their combined effect would be. The results were precision of 35.2%, recall of 26.9%, f-score of 30.5%, and accuracy of 86.7%. This is an improvement from the text features alone, mostly an improvement in recall. It is also an improvement in recall from the RST features alone, but the improvement in recall is accompanied by a decrease in precision.

Another way to explore the performance of the model is to look at whether units that should be selected are more probable than the units that should not be selected regardless of what those probability values are. To do this, for each text the units were ordered in terms of probability of being selected. The gold-standard summaries provide the correct number of units to select, x. This number of units, x, from the list of the most probable units were chosen and designated as selected. All other units were not selected. Performance was evaluated by calculating precision and recall using these labels. For comparison, x units were also randomly chosen as selected without considering their probabilities. This comparison shows whether the RST features, through their effect on probabilities, improve classification performance from randomly selecting the same number of units. The random version was run multiple times since the results may vary. The results are shown in the following table. Note that because the number of units to select is specified rather than determined by the classifier, the number of false positives and false negatives will be equal, causing precision and recall to be equal.

Model	Precision	Recall	F-score	Accuracy
Probability	0.327	0.327	0.327	0.854
Random 1	0.122	0.122	0.122	0.810
Random 2	0.138	0.138	0.138	0.813
Random 3	0.125	0.125	0.125	0.811
Random 4	0.113	0.113	0.113	0.808
Random 5	0.107	0.107	0.107	0.807

Table 3.2: Results of selecting units based on probability compared to randomly selecting the same number of units

These results show that RST information improves performance as measured by f-score by about 20% from a random baseline. When the model is told how many units to select for a summary and chooses them based on probabilities assigned by the classifier, it performs much better than randomly selecting the same number of units. Therefore, while the numbers may seem low overall, the RST features do have a positive effect on performance. More generally, these results provide support for using RST information for summarization.

4.3 Previous Work Using RST for Automatic Summarization

Previous research has also focused on how RST information can be used for summarization (Marcu 2000; Chengcheng 2010; Cardoso et al. 2015; Goyal and Eisenstein 2016). For example, Marcu (2000) explores how to use the nucleus/satellite distinction to decide which information is most important for a summary. He assumes that a text is represented as a tree where text units are leaf nodes and rhetorical relations are internal nodes that have children nodes representing the units that are the arguments of these relations. Each node is labeled as either a nucleus or satellite depending on its role in the relation. Each node is also associated with what he calls the *promotion set*, an important part of how Marcu uses discourse structure for summarization. The promotion set of a node is the set of units that represent the most important or salient elements of the

text span covered by that node. Salience is determined by the nucleus/satellite distinction, with nuclei considered more salient than satellites. For a leaf node, the promotion set is simply the node itself. The promotion set of a non-leaf node contains the node that is the nucleus of the text that it spans over. Thus, if a node has one nucleus child and one satellite child, the promotion set will contain the nucleus child. In this way, nuclei are promoted up through the tree as the most important units. Promotion sets can be seen in the following diagram.

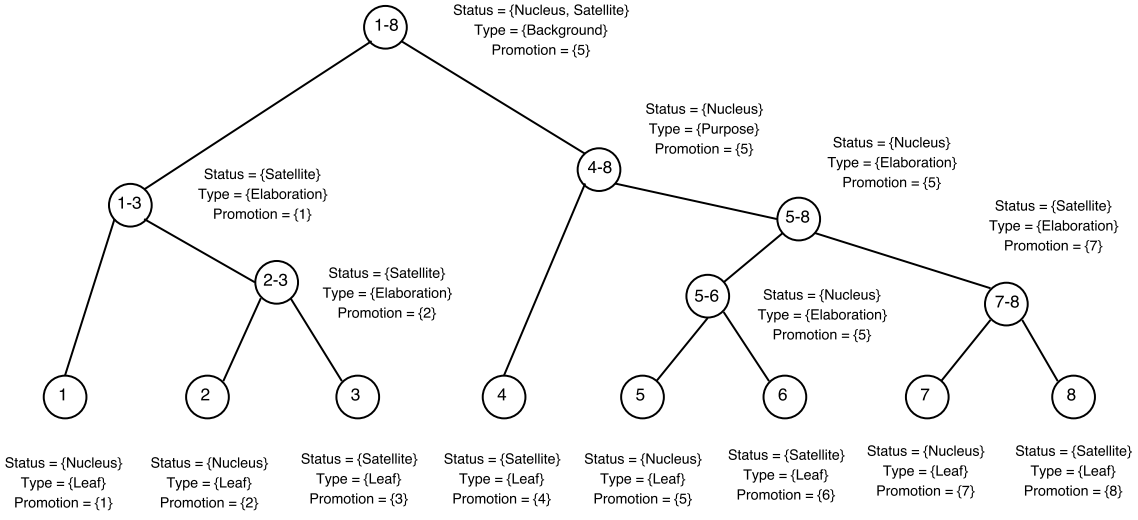


Figure 3.2: RST diagram illustrating promotion sets

For each leaf node, the promotion set consists of the node itself. As an example of the promotion set of an internal node, clauses 2 and 3 are connected by an elaboration relation. Clause 2 is the nucleus, and clause 3 is the satellite. For the span 2-3, the promotion set is {2} because 2 is the nucleus. The span 2-3 is itself the satellite of an elaboration relation, with clause 1 as the nucleus. The promotion set of 1-3 is {1} because 1 is the nucleus. At the highest level of the tree, 4-8 is the nucleus and 1-3 is the satellite of a background relation. The promotion set is {5}. Nuclei are considered the most

important information and are promoted through the tree so that the promotion set of the root represents the most important units from the text.

Marcu uses the promotion sets in the RST tree to create a partial ordering of the sentences in a text based on their importance. Each text unit can be scored based on how high it is promoted in the tree. Marcu uses the following formula to compute these scores, where u is a unit, D is the discourse structure, and d is the depth.

$$score(u, D, d) = \begin{cases} 0 & \text{if } D \text{ is NIL,} \\ d & \text{if } u \in promotion(D), \\ d-1 & \text{if } u \in parentheticals(D), \\ \max(score(u, leftChild(D), d-1), & \text{otherwise} \\ \quad score(u, rightChild(D), d-1)) & \end{cases}$$

The function *promotion(D)* returns the promotion set of the node*, parentheticals(D)* returns the parenthetical units of a node, which are units that are part of a larger unit and are usually separated from adjacent text by parentheses or dashes, and *leftChild(D)* and *rightChild(D)* return the left and right subtrees of each node. The tree given above has a depth of 5. Using this formula gives the following scores for each unit in that tree. For example, unit 5 is in the promotion set of the root so it receives a score of 5, which is the depth of the tree.

Unit	1	2	3	4	5	6	7	8
Score	4	3	2	3	5	1	2	1

Table 3.3: Importance scores by unit

These importance scores create a partial ordering of the units in the text.

5 > 1 > 2, 4 > 3, 7 > 6, 8

A summary can then be created by choosing sentences from this partial order. For example, a one-sentence summary would contain unit 5, and a two-sentence summary would contain units 5 and 1. This method is based on the intuition that the nuclei of a text's discourse structure represent a good summary of the text. In building a summary,

Marcu selects a specified percentage of units from this ordering for a summary. Units are chosen from the beginning of the ordering, so a summary will contain the first *n%* units. Marcu tested this method on a set of five articles that were manually annotated with discourse structure. The results showed that this method of determining importance and selecting summaries achieved recall of 56% and precision of 67%, which represents better performance than the classifier discussed above. However, the dataset for this evaluation was very small, at only five articles.

In other work on how discourse information can be used for summarization, Louis et al. (2010) explore the usefulness of different features, including discourse features, for selecting content in extractive summarization. Since discourse relations indicate connections between different parts of a text, Louis et al. look at the utility of discourse-based features including ones based on Marcu's (2000) idea of a promotion set. Among the discourse-based features, there are some that score text units based on how high in the discourse tree they are promoted, and others that penalize satellite units relative to nucleus units. They also consider non-discourse features including sentence length and whether a sentence is paragraph initial. Using these different types of features, they see which features correlate with sentences selected by people as important. Their results find that higher scores for RST features correlate with important sentences. In addition, the non-discourse features were also significant predictors, with longer sentences and sentences at the beginning of documents and paragraphs tending to be selected as important. In the context of a classification task using logistic regression to predict which sentences from input documents are important, the structure-based RST features perform

better than non-discourse features. These results suggest that structural information is useful for determining which information should be included in a summary.

As a different way of using RST, Li et al. (2016) consider the importance of Elementary Discourse Units (EDUs) from RST for summarization. Their work is based on the observation that human-written summaries tend to involve more changes from the original text, such as deleting parts of a sentence, rather than taking complete sentences from the text as is done in automatic extractive summarization. Therefore, they propose using units below the sentence level, EDUs, for extractive summarization. Choosing EDUs for a summary rather than sentences allows a summarizer to select only the most important parts of a sentence while discarding less important parts. When summary length is constrained, this ability is particularly important. Including a long sentence in a summary may waste space if all of its EDUs are not equally important. Working at the EDU level allows for more flexibility with choosing content for a summary. Li et al. found that EDUs aligned well with concepts found by people, suggesting that when summarizing, people may be working at the EDU level rather than the higher level of sentences or the lower level of words. They also found that using EDUs as the units of summarization resulted in better performance than sentences, particularly when summary size is small. The findings of Li et al. provide additional motivation for using RST information, in this case the notion of EDUs, for summarization.

4.4 Proposed Use of Topics Based on RST

As rhetorical and structural information has been shown to be useful for summarization in previous research, there is additional motivation for combining this information with the idea of topics. The research in this paper also focuses on combining

topic information with rhetorical structure. There are several places in a summarization system where rhetorical information could be incorporated, such as when dividing texts into topics or determining which sentences from a particular topic should be included in a summary. As I am interested in the impact of topic structure on summarization, I focus on using rhetorical information as part of the topic division process.

The main notion of topic that I am exploring uses information from RST, specifically RST topic relations. RST relations capture how parts of a text connect to each other to accomplish the writer's purpose. They therefore provide useful information about which sentences in a text are most closely related to each other. Grouping sentences according to how they are related in a rhetorical structure provides a way to divide texts into topics.

Mann and Thompson (1988) present a list of RST relations, but they suggest that it is an open list that can be modified to include other relations. Marcu (1999) proposes an extended set of relations. These relations were used to produce the RST Discourse Treebank (Carlson et al. 2001), a set of documents annotated with RST structure. This extended set of relations includes several relation types that capture higher-level relationships and organizational principles of text. Of particular importance to the current work are two relations related to topic changes within text. These types are topic shift and topic drift. Topic shift is a relation that connects large sections of text when there is an abrupt change between topics. The following example (Marcu 1999) shows a news article that discusses one news story in the first topic and a different story in the second topic.

[GAF TRIAL goes to round three. Attorneys in the third stock-manipulation trial of GAF Corp. began opening arguments yesterday in the Manhattan courtroom of U.S. District Judge Mary Johnson Lowe. In an eight-count indictment, the government has charged GAF, a Wayne, N.J., specialty chemical maker, and its Vice Chairman James T. Sherwin

with attempting to manipulate the common stock of Union Carbide Corp. in advance of GAF's planned sale of a large block of the stock in November 1986. The first two GAF trials ended in mistrials earlier this year. This trial is expected to last five weeks.]Topic 1
[SWITCHING TO THE DEFENSE: A former member of the prosecution team in the Iran/Contra affair joined the Chicago firm of Mayer, Brown & Platt. Michael R. Bromwich, a member since January 1987 of the three-lawyer trial team in the prosecution of Oliver North, became a partner in the Washington, D.C., office of the 520-lawyer firm. He will specialize in white-collar criminal defense work. Mr. Bromwich, 35, also has served as deputy chief and chief of the narcotics unit for the U.S. attorney's office for the Southern District of New York, based in Manhattan.]Topic 2

On the other hand, topic drift is a relation that connects large sections of text when the change between topics is smooth rather than abrupt, and there is still some similarity between topics. In the following example (Marcu 1999), there is a more subtle change in topic from "smart cards" in the first topic to "standards for smart cards" in the second topic.

[Smart cards are not a new phenomenon. They have been in development since the late 1970s and have found major applications in Europe, with more than a quarter of a billion cards made so far. The vast majority of chips have gone into prepaid, disposable telephone cards, but even so the experience gained has reduced manufacturing costs, improved reliability and proved the viability of smart cards.]Topic 1
[International and national standards for smart cards are well under development to ensure that cards, readers and the software for the many different applications that may reside on them can work together seamlessly and securely. Standards set by the International Organization for Standardization (ISO), for example, govern the placement of contacts on the face of a smart card so that any card and reader will be able to connect.]Topic 2

These topic relations connect large sections of text when building an RST structure, and they also provide a way to divide a text into smaller sections that share a common theme. Topics given by RST incorporate structural and semantic information without relying solely on word distribution. In contrast to notions of topic that operate at the sentence level, RST topics can span over multiple sentences and capture what larger sections of text are about.

This type of topic is expected to be useful for summarization. Since summarization involves greatly decreasing the number of sentences from the original text, it is necessary to generalize across sentences and find what they have in common. Topics that span multiple sentences provide a way of dividing texts into groups of related information. RST topic relations are defined to indicate changes in topic, and the examples above show that the resulting topics agree with the intuition that topics are sections of text whose sentences are about the same idea, have internal coherence, and share a common theme. In addition, the way RST relations are defined ensures that the adjacency of sentences in the original text is preserved in the RST structure. This agrees with how topics are likely structured in the mind of the person writing the text. When writing, people generally have topics in mind. Once a topic is introduced, it is described to completion before moving on to the next topic. Topics are typically discussed in adjacent sentences rather than a text skipping back and forth between topics. When reading a text, people also expect this type of structure with related sentences occurring next to each other in a group. The fact that RST topics contain adjacent sentences is a property that should make them useful for summarization because they capture a type of topic that is most salient for people writing and reading texts. Therefore, these topics are a promising approach for using topics for summarization, and the use of these topics will be explored in detail in the next chapter.

4.5 Exploration of RST Topics in Human-Written Summaries

Before experimenting with using RST topics in an automatic summarization system, I explored properties of these topics and their representation in manually-created summaries. Specifically, I looked at features such as how large topics are in terms of the

number of sentences they contain, how many topics are in a document, and how many sentences from a topic are in a summary. These properties give an overview of how this topic structure looks in actual documents and how these topics correlate with the choice of sentences for a summary. Understanding the connections between these topics and gold standard summaries gives a sense of how these topics will be useful for an automatic summarization system.

The data came from the RST Discourse Treebank (Carlson et al. 2002). Not all of the documents in the RST corpus contain topic shift or topic drift relations. These explorations focused on a subset of the training set of the corpus consisting of 40 documents with topic relations as I was specifically interested in how these relations are connected to the content of the gold standard summaries. To divide texts into topics, topic relations are used as dividing points to create a partition of a text into a sequence of non-overlapping topics so that all units in a text are included in one and only one topic. An in-depth description of this topic division method will be given in the next chapter.

Different texts contain different numbers of topics, and topics also vary in size. The following diagrams provide an overview of how these qualities vary in the dataset. They show the range of numbers of topics contained in a text as well as how many sentences are contained in a topic. The number of topics that is most common is two. Most documents have two, three, or four topics. Intuitively, these relatively small numbers of topics make sense as news articles generally discuss only a few topics. Looking at topic sizes, the number of sentences in a topic is likely to vary depending on the size of the entire document. In general, most topics contain between one and 20 sentences.

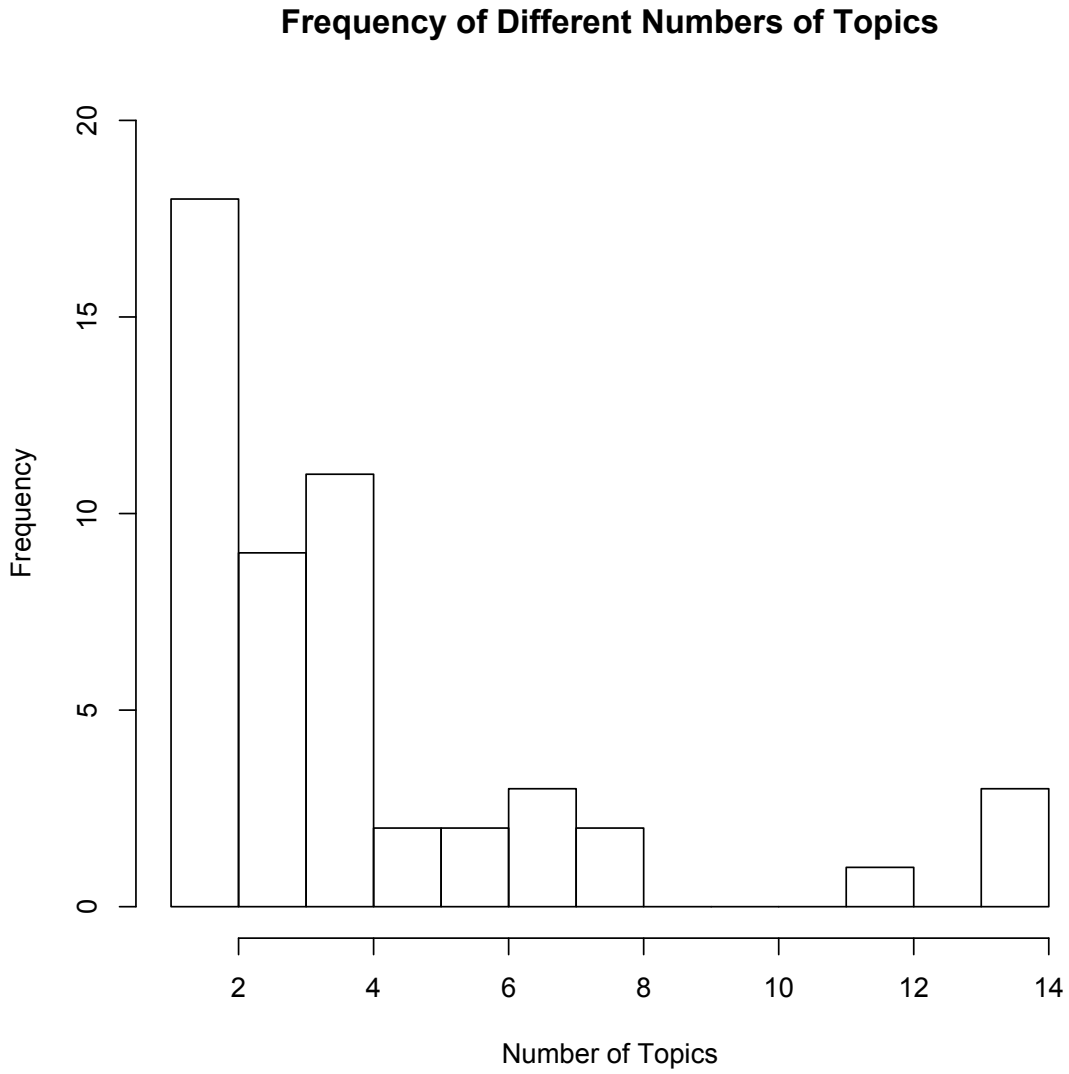


Figure 3.3: Number of texts containing different numbers of topics

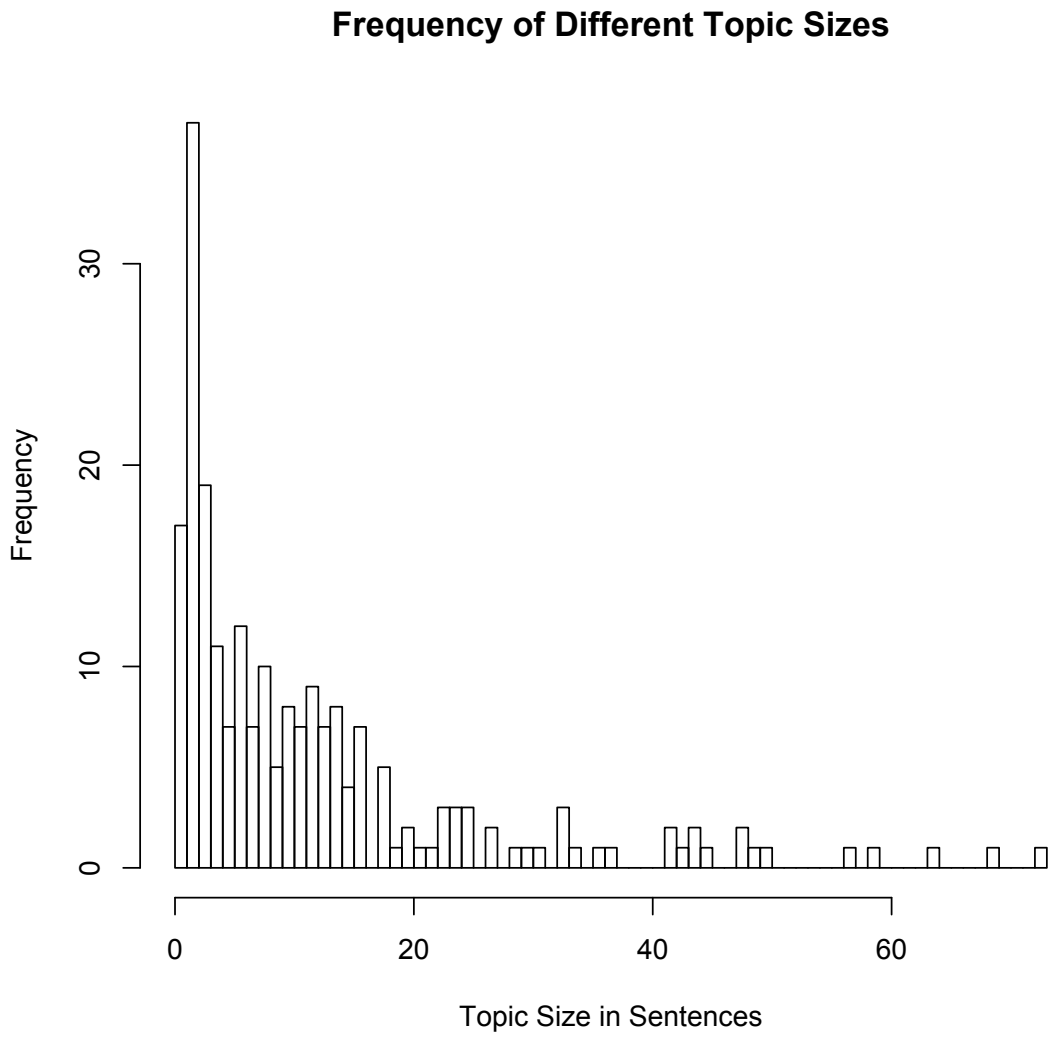


Figure 3.4: Frequency of different topic sizes

With the texts divided into these topics, I explored how different features such as topic size, summary size, and text size interact with whether or not a particular unit is selected for a human-created summary. The following plots show these interactions.

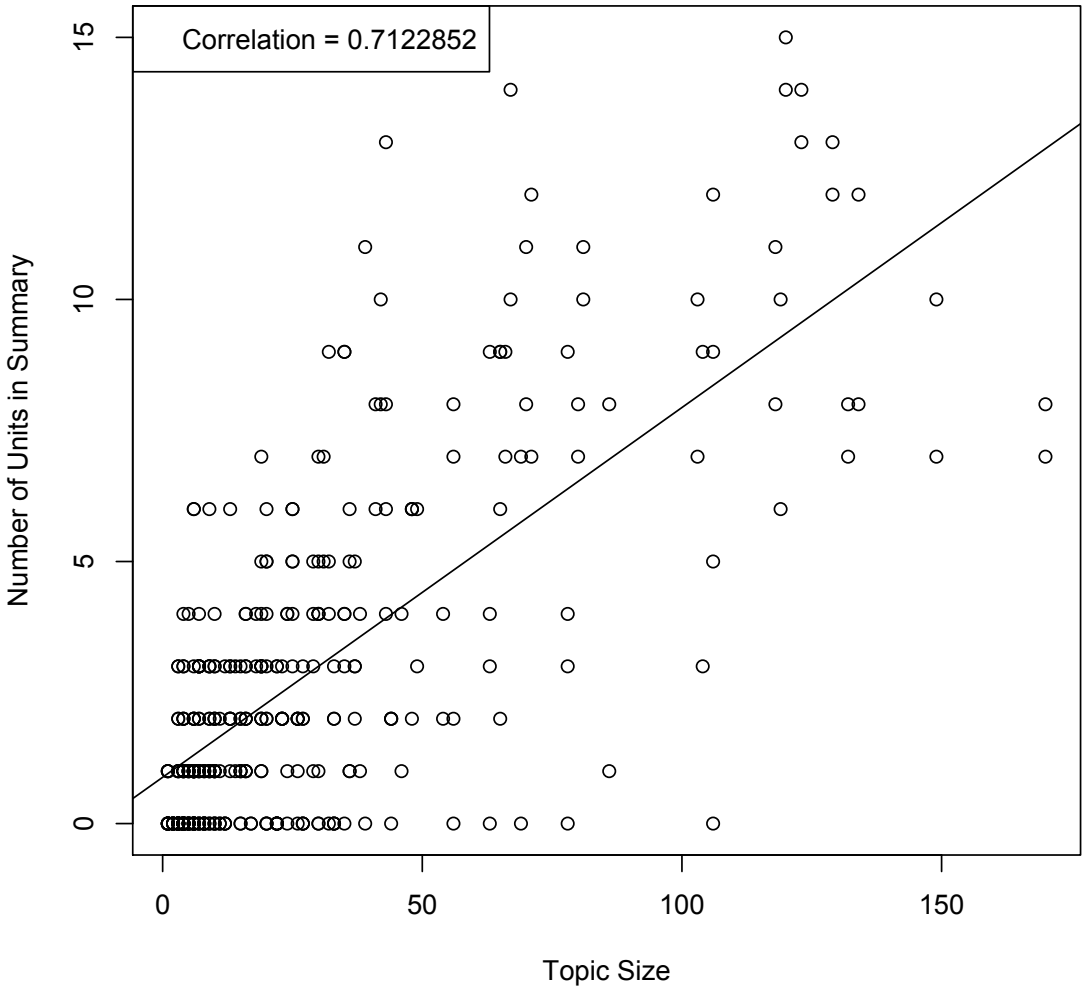


Figure 3.5: Topic size compared to the count of units from topic in the summary

This plot shows the number of units appearing in a summary from a particular topic on the y-axis compared to the size of that topic, the total number of units in that topic, on the x-axis. As topics get larger, they tend to have more units in a summary. This is expected since larger topics have more units to choose from.

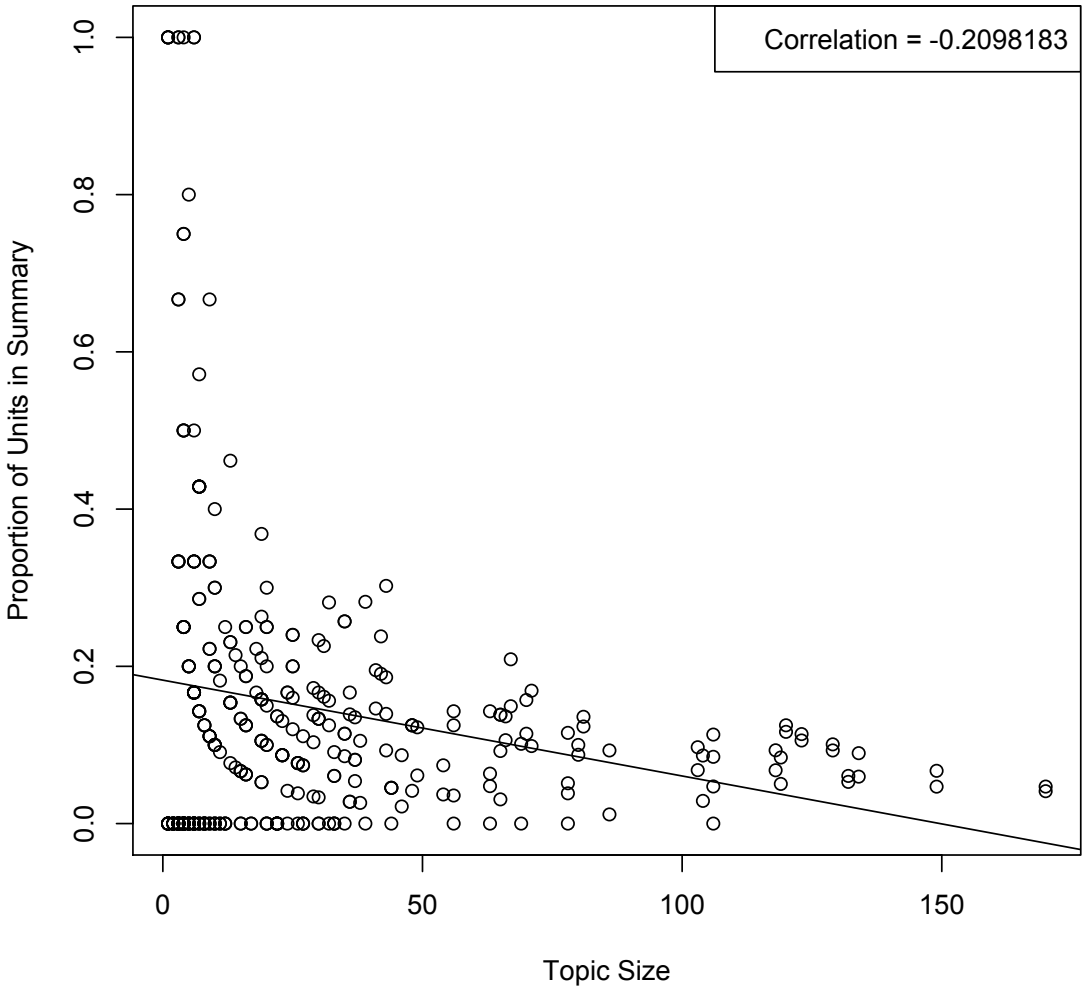


Figure 3.6: Size of topic compared to proportion of units from topic in summary

In this plot, the x-axis is the same as in the previous plot. The y-axis is now the proportion of units in a topic that appear in a summary rather than the raw count of units. There is a slight negative correlation between how large a topic is and how much of that topic is included in a summary. Smaller topics range from being completely represented in a summary to not being included at all. For example, if a topic contains a single unit, that topic will either be 0% or 100% included in a summary. As topics get larger, the

proportion of their units that appear in a summary gets smaller. This is evidence of pressure to include different topics in a summary. Intuitively, for large topics, there is less benefit derived from including all of that topic's units in a summary compared to including units from a different topic. At a certain point, the information from a topic will have been conveyed by the units already selected from that topic and adding additional units from that topic is unlikely to improve the summary. If the number of units from a topic in a summary were completely proportional to the size of the topic, the line should be horizontal, as the number of units in the summary would increase steadily as the topic size increases. The negative slope and correlation indicate that there is some pressure to include units from different topics, and smaller topics may receive proportionally more representation in a summary than larger topics.

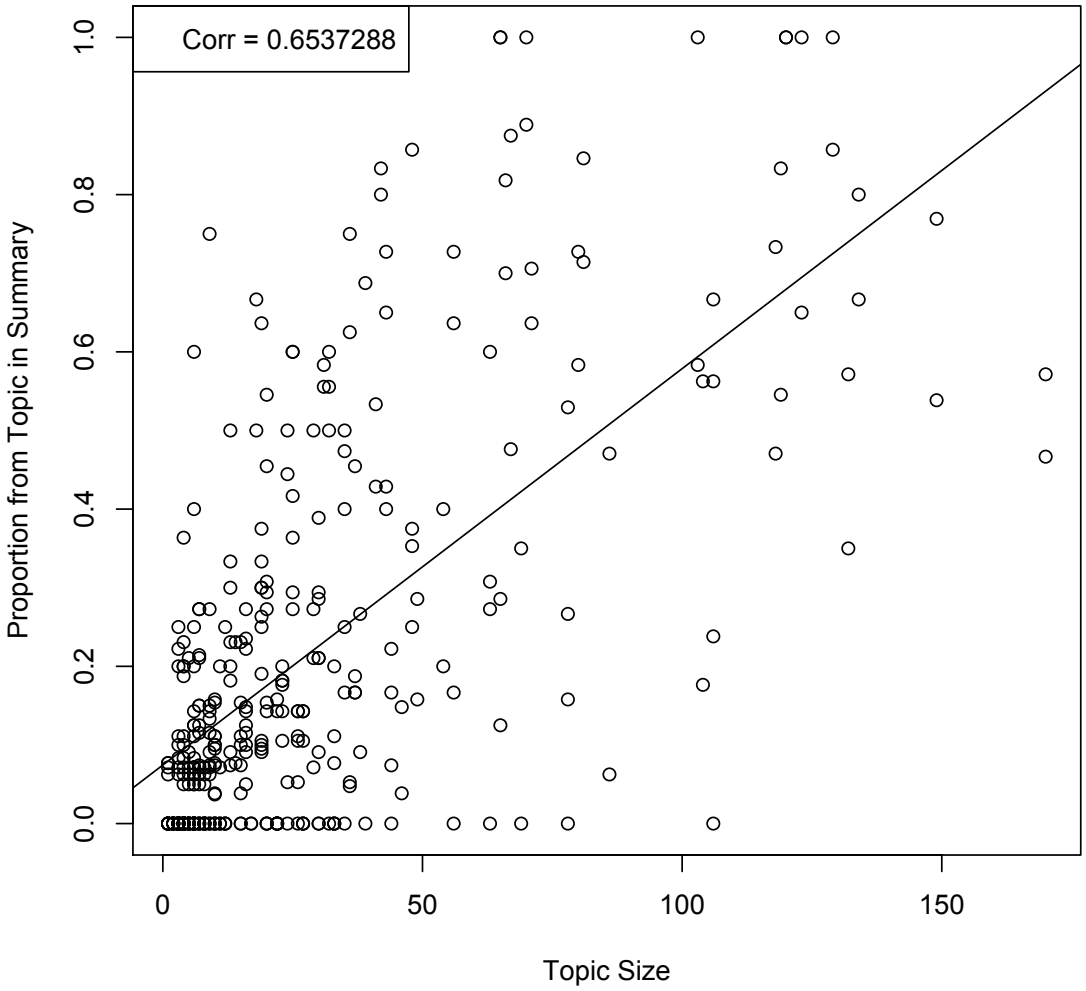


Figure 3.7: Size of topic compared to proportion of summary from that topic

While the last plot looked at the proportion of a topic that appears in a summary, this plot looks at the proportion of a summary that comes from a particular topic. The denominator for the ratio on the y-axis is the summary size rather than the topic size. The x-axis is again the size of the topic. This plot explores the question of whether all topics are represented equally in a summary regardless of size or whether the size of the topic influences how much of the summary comes from that topic. On the one hand, equal representation ensures that all topics are covered in the summary. On the other hand, if

one topic is much larger than another, it is possible that a good summary should include more information from the larger topic in order to convey the same information as the original text.

Overall, there appears to be a fairly strong positive correlation with larger topics making up a larger portion of a summary. The line should be horizontal if all topics received the same amount of coverage in a summary. While there is an overall positive correlation, looking more closely, there is a cluster of points with topic size less than 50 where there seems to be a correlation, but beyond 50, the point distribution appears fairly random. In order to better understand the correlation, the data was divided into smaller groups by topic size. The results are shown below.

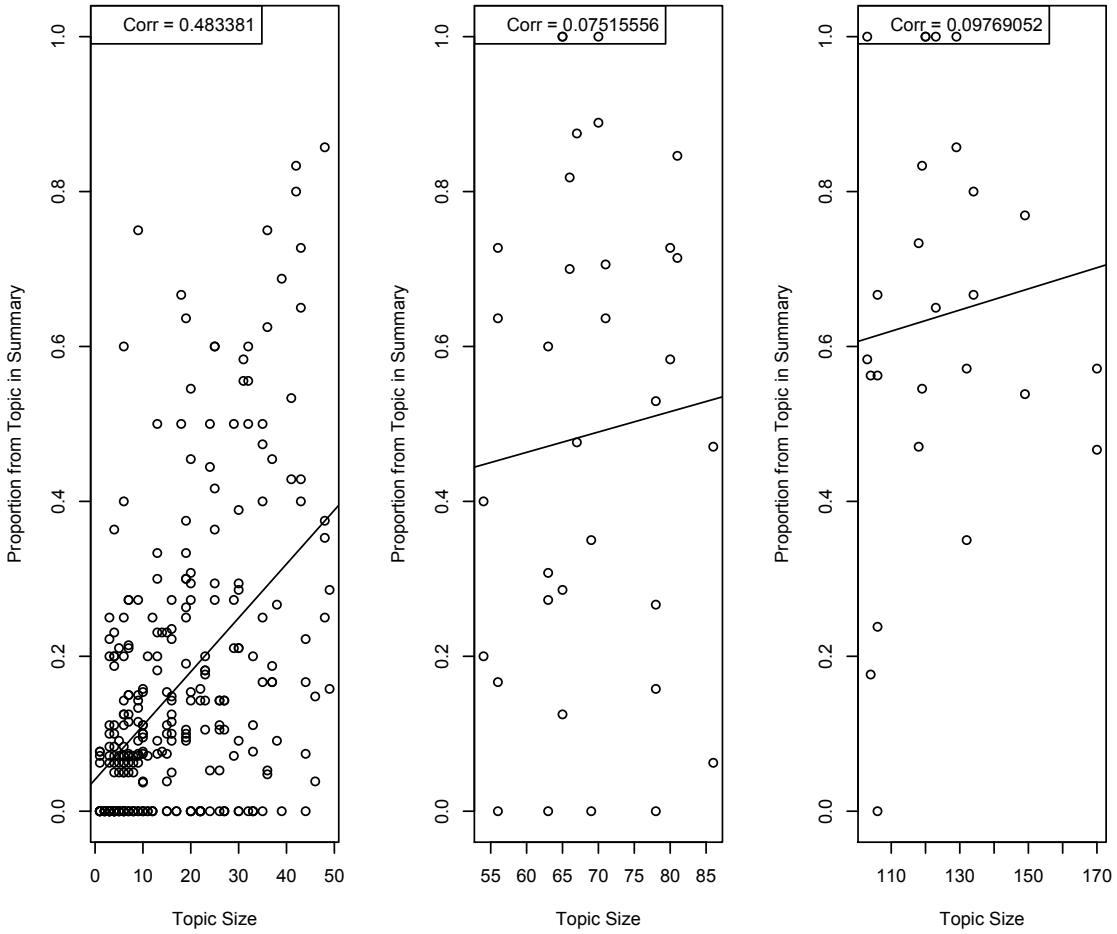


Figure 3.8: Results broken down by topic size

When the plot is broken down in this way, there are clear differences in the correlation depending on topic size. As the original plot suggested, once the topic size is larger than 50, there is no significant correlation between the size and how much of the summary comes from that topic. For smaller topics, there is a correlation with the amount of representation in a summary increasing as topic size increases. Similar to the previous plots, this series of plots shows that the amount of a summary that comes from a topic increases as the topic gets larger, but only up to a certain point. When a topic becomes

large enough, it no longer receives coverage in proportion to its size, suggesting that additional coverage may not be beneficial. The lack of proportional coverage at larger sizes may leave room for coverage of smaller topics, which would otherwise not be included in a summary at all.

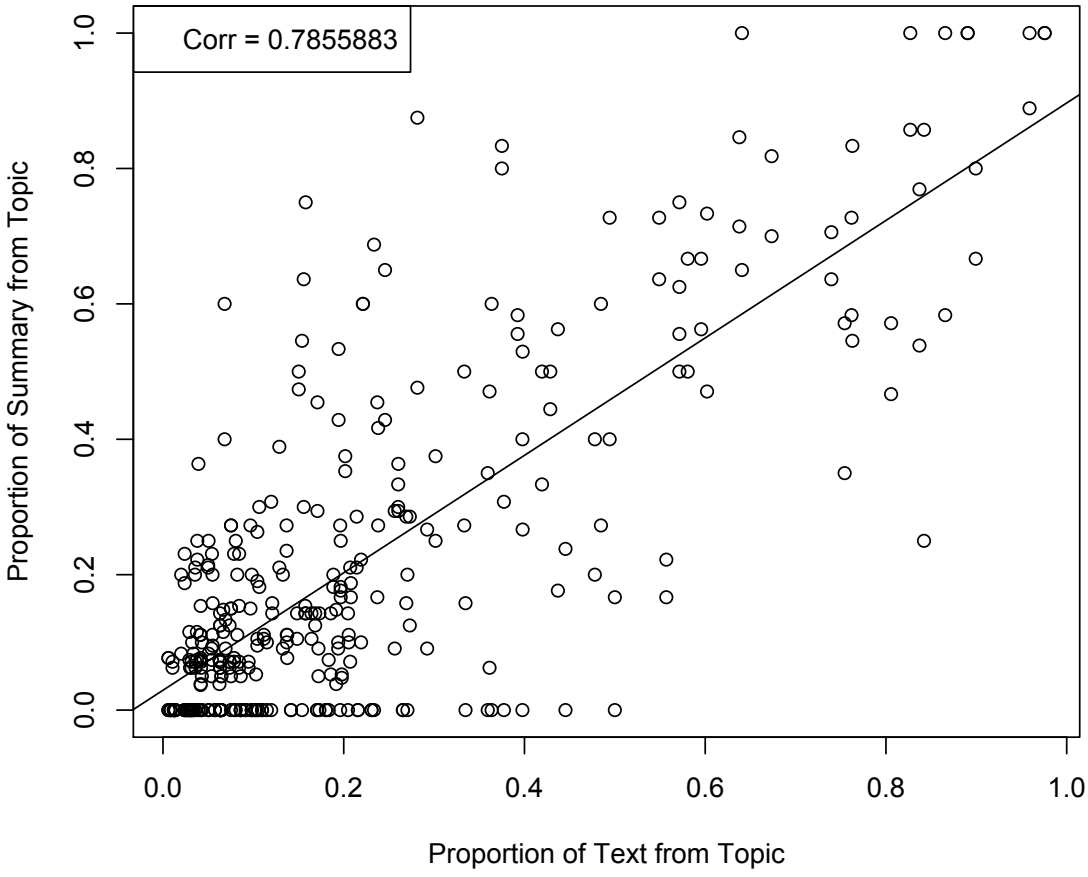


Figure 3.9: Proportion of summary compared to text from topic

An important comparison to consider is how the proportion of units in a summary from a particular topic is related to the proportion of the original text that is made up of that topic. The baseline unconditioned probability of a unit being in a summary is the ratio of the size of the summary to the size of the text. If there is no influence of topic structure, it would be expected that whether or not a unit is included in a summary depends entirely on how large the summary is relative to the text. If a summary is 25% of the size of the original text in terms of number of units, then each unit has a 25% chance of being in the summary. Therefore, if that were the case, the slope of the linear regression line should be 1 in the above plot. As a topic becomes a bigger proportion of the text, that topic should also become a bigger proportion of the summary at the same rate. The plot shows that the slope is less than 1. This means that topics do not get the representation in a summary that would be expected based simply on how much of the text is part of that topic. This provides some evidence that topic structure plays a role in which units are selected for a summary. In general, it seems that the effect of topic structure is to shift some of the expected coverage of larger topics to smaller topics, ensuring that more topics are represented and creating slightly more equal representation of topics.

In order to confirm the effect of topic structure, a logistic regression looked at which factors have a significant influence on which units from a text are selected for a summary. The factors tested were topic size, text size, summary size, ratio of summary size to text size, and ratio of topic size to text size. Topic size, text size, and summary size are expressed as counts of EDUs. The ratios of summary size to text size and topic size to text size are expressed as numbers between 0 and 100. The goal is to see if any

factors other than the ratio of the summary size to the text size have an effect, as that factor represents a baseline equivalent to randomly choosing units from the text. The results are presented in the following table, after normalizing values for each variable.

Factor	Estimate	P-value
Topic Size	-0.30650	0.00457 **
Text Size	0.02192	0.87401
Summary Size	0.01566	0.87089
Summary to Text Ratio	0.17171	0.02804 *
Topic to Text Ratio	0.10120	0.29010

Table 3.4: Logistic regression results testing factors influencing summary selection

Two factors were significant. As expected, the baseline ratio of the size of the summary to the size of the text had a significant effect on whether a unit is chosen for a summary. However, the results also showed that topic size was a significant factor. This provides evidence that topic structure has an effect on which information is included in a summary.

The explorations in this section suggest that topic structure does provide information that is useful for summarization. The probability of a unit of text being included in a summary not only depends on the size of the summary but also on the topic it is contained in. In general, the texts in this dataset contain a relatively small number of topics with various numbers of sentences in each topic. As topic structure has been shown to correlate with which information is included in human-written summaries, it will be interesting to explore how topics can be used when automatically summarizing.

4.6 RST Topics Compared to Topic Segmentations

To understand how the topics produced by RST compare to the divisions produced by text segmentation methods, RST topics were compared to the output of two text segmentation algorithms, C99 (Choi 2004) and TextTiling (Hearst 1997). Both of

these models are described earlier in this chapter. Implementations of these algorithms come from a Python package for unsupervised text segmentation[2].

Several evaluation measures for text segmentation were used to compare the segmentations produced by C99 and TextTiling to the RST segmentation. One of these measures, P_k (Beeferman et al. 1999), compares a proposed segmentation to a reference segmentation and calculates the probability that two sentences a distance of k sentences apart are not classified consistently in both segmentations. In other words, it calculates the probability that two sentences occur in different segments in one segmentation and in the same segment in the other segmentation. The value of k is set to half of the average true segment size. Penalties are calculated by moving a window of size k through the text and determining whether the two ends of the window are in the same or different segments in the hypothesis and reference segmentations. The number of disagreements is counted and divided by the total number of measurements. This measure calculates a penalty, and therefore a proposed segmentation that perfectly agrees with the reference segmentation receives a score of 0. Another evaluation measure, $WindowDiff$ (Pevzner and Hearst 2002), similarly uses a moving window but instead of checking whether sentences occur in the same segments, it compares the number of boundaries that occur within the window in the hypothesized and reference segmentations. A penalty is incurred when the two segmentations contain different numbers of boundaries. Compared to other evaluation measures such as precision and recall, these measures using sliding windows are able to penalize segmentations for containing different boundaries while taking into account the degree of difference so that near misses are not penalized as

[2] https://github.com/intfloat/uts

much. Implementations of these evaluation measures come from the SegEval Python package for text segmentation evaluation (Fournier 2013).

The results of the comparisons are presented in the following table. Calculating both P_k and $WindowDiff$ involves specifying one segmentation as the hypothesis and the other as the reference. The window size used in the calculation of these measures depends on the size of the segments in the reference segmentation. None of these segmentations are a gold standard segmentation produced by people performing the task of segmentation. Therefore, the results were calculated with both the RST topics and the text segmentation output serving as the reference. All results are presented below.

	Average P_k	Average WindowDiff
C99 (RST as reference)	0.572	0.725
RST (C99 as reference)	0.495	0.512
TextTiling (RST as reference)	0.622	0.717
RST (TextTiling as reference)	0.458	0.470

Table 3.5: Results of comparing RST topics to other segmentations

The results show that there tends to be a lot of disagreement between the RST topic segmentation and the segmentations from the other algorithms, as larger values indicate more disagreement. Looking at a few examples provides more information about the differences in the segmentations. The following examples show the sizes of the segments in each segmentation.

Segmentation according to C99	Segmentation according to RST
(3, 7, 3, 7, 4, 3, 8, 6, 3, 9, 7, 4, 9, 2, 5, 3, 6, 3, 2)	(27, 25, 42)
(16, 56, 10, 16)	(25, 73)
(2, 4, 3, 2, 3, 2, 4, 4, 4, 3, 4, 3, 3, 4, 5)	(31, 9, 10)
(4, 4, 4, 9, 3, 12, 2, 5, 4)	(8, 4, 10, 5, 11, 9)
(8, 25, 3)	(15, 15, 3, 3)

Table 3.6: Examples of differing segmentations

As seen in the examples, the C99 algorithm tends to place more boundaries and divide texts into more total segments than RST. Because more segments are found by C99, the

segments also tend to be smaller, with many containing only a few sentences. While the number of boundaries clearly differs between segmentations, another factor to consider is whether the boundaries that are found in both segmentations line up. For example, in the first example, both methods place a boundary after 27 units, even though C99 also places additional boundaries. However, in the second example, none of the boundaries agree between the two segmentations. The results show agreement in some cases but not in others.

Although the method for topic division using RST topics is similar to the text segmentation algorithms by producing a sequence of non-overlapping segments, the RST topics differ from the segmentations in terms of number of segments and size of segments.

5 Conclusion

This chapter described several ideas of what it means to be a topic and how to determine topics automatically, including topics at different levels of granularity and topics determined using different types of information. It also discussed motivation for using topic structure for summarization, which includes studies of how people compose and process text and a clear link between the types of text relationships that are captured by topics and the relationships that are useful for determining summary content. The notion of RST topics was introduced. These topics represent high-level structural relationships between sections of text, and they align with the intuitive definition of topics as coherent parts of text with a common theme. Additional motivation for the use of topic structure for summarization was given by explorations of how the topics from RST are represented in summaries created by people. A logistic regression showed that

factors related to topic structure influence which information is selected for inclusion in summaries. Given this discussion of topics, and RST topics in particular, the next chapter will describe how these topics can be incorporated into a summarization system.

Chapter 4

Experiments using Topics for Summarization

1 Overview

The previous chapter described notions of what it means to be a topic based on different types of information. The main notion of topic under consideration is one based on relations from Rhetorical Structure Theory. Another notion of topic comes from common topic modeling methods. The goal of exploring these different definitions of topic is to use topic structure to improve performance on the task of automatic summarization. The previous chapters motivated the use of topic structure as a way to produce summaries that include coverage of all main ideas in a document and therefore convey the same information as the original document.

Given the discussion of topics and why they are expected to be useful for summarization, this chapter describes the experiments performed to test the use of topics for summarization. In order to see how useful these different notions of topic are, they were incorporated into a summarization system. To compare the effects of using topics versus not using topics, summarization was either performed at the level of the whole text or at the level of individual topics. Specifically, the process for incorporating topics into summarization included the following steps: divide a text into topics, summarize the text of each topic, and concatenate the summaries of each topic to create a summary for the whole text. With this method, topics are treated as independent pieces of text that

contribute to the overall meaning of the text, and each topic will be represented in the final summary. This agrees with the intuition that texts can be divided into topics and a good summary should contain coverage of all topics that appear in the original text. This is a straightforward way to see how topics affect summarization. The following sections describe the methods for performing and evaluating summarization.

Section 2 explains the methods that were used, including how to divide a text into topics and how summarization was performed. Section 3 presents the main results of using RST-based topics for summarization. In Section 4, experiments and results with a different notion of topic based on Latent Semantic Analysis are described. Section 5 explores the possibility of extending the use of RST topics for summarization to unannotated documents using an automatic RST parser. Section 6 returns to the relationship between summarization and compression with an experiment considering how similar or dissimilar different topics are. Section 7 summarizes the main findings and concludes the chapter.

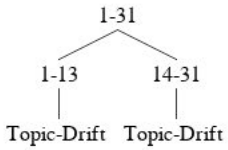
2 Methods

2.1 Dividing a Text into Topics

The main notion of topic under consideration is based on RST. Topic shift and topic drift relations are RST annotations that indicate changes in topic. Given those relations, there is still a question of exactly how to use them to divide a text into topics. This section describes a method for dividing texts into topics using RST topic relations so that every EDU is included in one and only one topic. In the most straightforward case, all units in the text are explicitly designated as part of a topic. This case can be seen in the example RST annotation below. The first line indicates that the text contains 31 units, as

the Root spans the entire text. The next line shows that units 1-13 are part of a topic-drift relation. Skipping down to the other argument of this relation on the last line shows that the other element of the topic relation includes the rest of the text, units 14-31. In this notation, these text spans are arguments of the same relation because when combined they form a continuous sequence with the second argument starting directly after the first, and visually the two arguments occur at the same indent level. Therefore, this text can easily be divided into two topics. The first topic begins with the first unit of the text and continues to unit 13, and the second topic begins at unit 14 and continues to the end of the text.

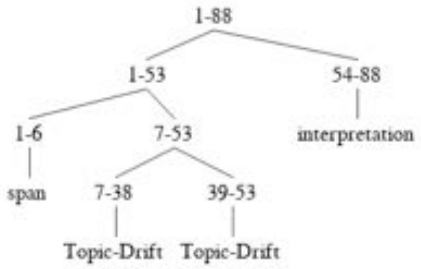
(Root (span 1 31)
  (Nucleus (span 1 13) (rel2par **Topic-Drift**)
    (Nucleus (span 1 8) (rel2par span)
        .
        .
        .
  (Nucleus (span 14 31) (rel2par **Topic-Drift**)



However, in other texts, not all units within a text are necessarily included as part of an explicit topic relation. In these cases, in order to divide a text into topics, the topic relations are used as dividing points. Each occurrence of a topic relation signaled the beginning of a new topic. Anything before that point is grouped together as a topic, and anything after a topic relation is grouped as a topic. In that way, all units in a text are included as part of a topic. For example, if a topic relation started at unit 5, units 1-4 would be considered a topic. This topic division can be seen in the following example. The first line shows that the text contains 88 units. In contrast to the previous example,

the first relation is not a topic relation. The first explicit topic relation begins with unit 7 and ends with unit 38. The other argument of that relation begins with unit 39 and continues to unit 53. No explicit topic relations include either the beginning or the end of the text. Using the topic relations as dividing points, all units can be placed into a topic. Units 1-6 become a topic, spanning from the beginning of the text to the first topic relation, and units 54-88 become a topic, spanning from the end of a topic relation to the end of the text.

(Root (span 1 88)
  (Nucleus (span 1 53) (rel2par span)
    (Nucleus (span 1 6) (rel2par span)
      (Satellite (span 1 4) (rel2par concession)
        (Nucleus (leaf 1) (rel2par span) (text))
        (Satellite (span 2 4) (rel2par elaboration-object-attribute-e)
          (Satellite (span 2 3) (rel2par means)
            (Satellite (leaf 2) (rel2par attribution) (text))
            (Nucleus (leaf 3) (rel2par span) (text))
)
          (Nucleus (leaf 4) (rel2par span) (text))
)
)
      (Nucleus (span 5 6) (rel2par span)
        (Satellite (leaf 5) (rel2par attribution) (text))
        (Nucleus (leaf 6) (rel2par span) (text))
)
)
    (Satellite (span 7 53) (rel2par background)
      (Nucleus (span 7 38) (rel2par **Topic-Drift**)
          .
          .
      (Nucleus (span 39 53) (rel2par **Topic-Drift**)
        (Satellite (span 39 42) (rel2par background)
          .
          .
(Satellite (span 84 88) (rel2par elaboration-additional)



This division method creates a partition of the text into a sequence of non-overlapping topics. Using this method means that topics contain adjacent units, and each unit in the text is contained in exactly one topic. The division is deterministic with only one possible division into topics for each text. This method is used to divide all texts in the dataset into their topics. Pseudocode for the topic division process is shown in the following figure.

```
input: RST annotation file
Topics = list of topics

# Find explicit topics in RST annotations
(1)  for line in annotations:
(2)       relation = find label of relation type
(3)       span = (x, y) where x is starting unit and y is ending unit
(4)       if relation = 'topic-shift' or 'topic-drift':
(5)                 add span to Topics

# If there is not a topic starting with the first unit, add one
(6) minimum = lowest value in Topics
(7) if minimum != 1:
(8)       add (1, minimum-1) to Topics

# If there is not a topic ending with the last unit, add one
(9)   total_len = total number of units
(10) maximum = highest value in Topics
(11) if maximum < total_len:
(12)      add (maximum+1, total_len) to Topics

# Remove topics with overlapping starting or ending points to ensure sequence of
non-overlapping topics
(13) for (x, y) in Topics:
(14)      if y is not smallest value for x:
(15)                remove (x, y) from topics
(16)      if there are multiple values of x for y:
(17)                if x = lowest value:
(18)                          remove (x, y) from topics
(19)                          add to Topics (lowest value, w-1) where (w, z) in Topics and
w-lowest value is smallest

# If a unit is not included in any topic, add it as its own topic
(20) for i from 1 to total_len:
(21)      if i is not covered by any topic in Topics:
(22)                add (i, i) to Topics

(23) return Topics
```

Figure 4.1: Pseudocode for dividing text into topics

The following are two example texts that are divided into topics according to the topic relations in their RST annotations. In this case, each text has two topics. In both texts, it is possible to notice a change in topic between the two sections. In the first example, the first topic discusses Nissan's earnings, while the second topic describes Nissan's production plans. The second text is about the FDA's interaction with two pharmaceutical companies, and each topic discusses one of these companies. These examples illustrate the types of topics that are captured by RST relations.

Example Text 1 with Topics

> [Nissan Motor Co. expects net income to reach 120 billion yen (U.S. $857 million) in its current fiscal year, up from 114.6 billion yen in the previous year, Yutaka Kume, president, said. Mr. Kume made the earnings projection for fiscal 1990, ending next March 31, in an interview with U.S. automotive writers attending the Tokyo Motor Show. The executive said that the anticipated earnings increase is fairly modest because Nissan is spending heavily to bolster its dealership network in Japan and because of currency-exchange fluctuations.]Topic 1
>
> [During the next decade, Mr. Kume said, Nissan plans to boost overseas vehicle production sufficiently to account for a majority of sales outside Japan.
> Last year, Mr. Kume said, Nissan exported slightly over one million vehicles, and produced 570,000 cars and trucks at its factories in North America, Europe and Australia. But by 1992, he added, Nissan will build one million vehicles a year outside Japan, or sufficient to equal exports. "By the end of the 1990s," he said, "we want to be producing roughly two vehicles overseas for every vehicle that we export from Japan." That will involve a substantial increase in overseas manufacturing capacity, he acknowledged, but didn't provide specific details.]Topic 2

Example Text 2 with Topics

> [Food and Drug Administration spokesman Jeff Nesbit said the agency has turned over evidence in a criminal investigation concerning Vitarine Pharmaceuticals Inc. to the U.S. Attorney's office in Baltimore. Neither Vitarine nor any of the Springfield Gardens, N.Y., company's officials or employees have been charged with any crimes. Vitarine won approval to market a version of a blood pressure medicine but acknowledged that it substituted a SmithKline Beecham PLC product as its own in tests.]Topic 1

[Mr. Nesbit also said the FDA has asked Bolar Pharmaceutical Co. to recall at the retail level its urinary tract antibiotic. But so far the company hasn't complied with that request, the spokesman said. Bolar, the subject of a criminal investigation by the FDA and the Inspector General's office of the Health and Human Services Department, only agreed to recall two strengths of its version of Macrodantin "as far down as direct customers, mostly wholesalers," Mr. Nesbit said. Bolar, of Copiague, N.Y., earlier began a voluntary recall of both its 100 milligram and 50 milligram versions of the drug. The FDA has said it presented evidence it uncovered to the company indicating that Bolar substituted the brand-name product for its own to gain government approval to sell generic versions of Macrodantin. Bolar has denied that it switched the brand-name product for its own in such testing.]Topic 2

2.2 Data

The data for these topic summarization experiments comes from the RST Discourse Treebank (Carlson et al. 2002). This corpus contains 385 Wall Street Journal articles that have been annotated with RST structure. This is the most commonly used dataset for experiments involving RST. Annotation was performed by trained annotators with previous experience in data annotation and language analysis (Carlson et al. 2003). Before annotation, they were trained in Rhetorical Structure Theory and participated in several practice sessions. Inter-annotator agreement was monitored and methods for improving agreement were discussed. Annotators followed a detailed set of guidelines, which can be found in the annotation manual (Carlson and Marcu 2001).

Dividing a text into RST topics depends on the presence of topic relations in the annotated text, specifically topic-shift or topic-drift relations. Not all texts in the corpus include topic relations in their annotations. Therefore, these experiments were limited to texts that do contain topic relations in order to directly compare how using topics affects performance compared to not using topics. In the corpus, there are 71 documents with topics.

Another feature of the RST Discourse Treebank is the presence of summaries for some documents. Gold-standard summaries are crucial for evaluating the output of a summarization system. For 150 documents in the corpus, there are 2 manually-created extractive summaries. Two analysts created these extracts by selecting a number of Elementary Discourse Units (EDUs) based on the square root of the total number of EDUs in the text. The following example is one of these extractive summaries. It includes the entire text, with one EDU per line. The EDUs selected by one analyst for the extractive summary are indicated with '*', and the EDUs selected by the other analyst are indicated with '#'.

```
*# Nissan Motor Co. expects net income to reach 120 billion yen
  (U.S. $857 million)
*# in its current fiscal year, up from 114.6 billion yen in the previous year,
*  Yutaka Kume, president, said.
  Mr. Kume made the earnings projection for fiscal 1990,
  ending next March 31,
  in an interview with U.S. automotive writers
  attending the Tokyo Motor Show.
  The executive said
  that the anticipated earnings increase is fairly modest
  because Nissan is spending heavily
  to bolster its dealership network in Japan
  and because of currency-exchange fluctuations.
*# During the next decade,
  Mr. Kume said,
*# Nissan plans to boost overseas vehicle production sufficiently
*  to account for a majority of sales outside Japan.
  Last year,
  Mr. Kume said,
  Nissan exported slightly over one million vehicles,
  and produced 570,000 cars and trucks at its factories in North America, Europe
        and Australia.
  But by 1992,
  he added,
 # Nissan will build one million vehicles a year outside Japan, or sufficient to
        equal exports.
  "By the end of the 1990s,"
  he said,
```

"we want to be producing roughly two vehicles overseas for every vehicle that we export from Japan."
That will involve a substantial increase in overseas manufacturing capacity, he acknowledged,
but didn't provide specific details.

There is some overlap, but not complete overlap between the units selected by different analysts, indicating the variation in human opinions of which information should be included in a summary, an issue discussed in previous chapters.

Since gold-standard summaries are required to evaluate system-produced summaries, these experiments were performed on texts that have corresponding summaries. Of the 71 documents in the corpus that have topics, 51 documents also have extractive summaries. These 51 documents are the core dataset for the topic summarization experiments. These documents include the summaries necessary for evaluation as well as the annotated topic relations necessary to evaluate the use of topics for summarization.

2.3 Evaluation

Summary evaluation is a difficult task. There can be more than one good summary of a text, and when people are instructed to create summaries, they do not necessarily contain the same sentences. While there is no single correct answer for what a summary should contain, evaluation typically involves comparing a system-produced summary to a manually-created reference summary. Summary quality is based on some measure of similarity or overlap with a reference summary. Since evaluation is a difficult problem, several different performance measures are used to capture different aspects of summary quality.

Evaluation measures for summarization tend to focus on finding superficial similarity. For example, they consider whether two summaries contain the same words. Whether a summary is a good representation of the information in the original text is obviously a more complex question than whether the two texts contain the same words. The same difficulties of determining whether sentences contain the same information that affect content selection for a summary also affect summary evaluation. However, with extractive summarization, the impact of this issue is reduced. In this type of summarization, summaries are created by taking pieces of text directly from the original text. Therefore, the produced summaries will contain the same words as the original text. If the reference summaries are also extractive, they will also contain words drawn directly from the original. Comparing the produced summaries to the reference summaries allows for a fair comparison of whether they contain the same information, even using superficial measures. While in general it would be beneficial to evaluate summaries using measures that go beyond surface similarity, such measures are harder to define and implement. As the automatically produced summaries and the reference summaries are both extractive in this work, these previously proposed measures that look at word overlap will be used.

ROUGE (Lin 2004) is a measure to evaluate performance on the task of automatic summarization. Recall-Oriented Understudy for Gisting Evaluation involves comparing a summary produced by a summarization system to reference or gold-standard summaries created by humans. Specifically, ROUGE-N measures n-gram (unigram, bigram, etc.) recall between a system summary and a reference summary. Recall refers to how many of

the reference n-grams were included in the system summary. The formal equation for ROUGE-N is presented below.

$$(4.1)\ ROUGE - N = \frac{\sum_{S \in \{Reference\ Summaries\}} \sum_{gram\ n \in S} Count_{match}(gram_n)}{\sum_{S \in \{Reference\ Summaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

In this equation, n refers to the size of the n-gram, such as unigram (1) or bigram (2). An n-gram itself is represented by $gram_n$, and $Count_{match}(gram_n)$ refers to the number of times that the n-gram $gram_n$ appears in the system summary. Therefore, the numerator is the number of matching n-grams, and the denominator is the total number of n-grams in the reference summaries.

ROUGE is a standard measure used in the field of summarization (Erkan and Radev 2004; Lin and Hovy 2003; Xie et al. 2008; Wong et al. 2008; Nallapati et al. 2016; Chopra et al. 2016). One downside of ROUGE is that it is entirely recall-based. In general, a summary will be rewarded for including more n-grams without being penalized for containing n-grams that do not appear in the reference summary. In the extreme case, a summary that is the same length as the original text being summarized could achieve perfect recall even though such a summary would clearly not be considered a good summary, since the goal of summarization is to produce a shortened version of the input. In order to avoid this problem, summary length must be controlled. Specifically, since ROUGE-N is a word-based evaluation measure, the summary length in terms of word count must be controlled so that system-produced summaries are similar in length to the reference summaries. Another way to reduce this problem is to also consider measures of precision that reward summaries for containing the same units as the references while penalizing them for including units that are not in the reference summaries. The other two

evaluation measures used do have this property, so they are less affected by differences in length.

Unit overlap is another evaluation measure for summarization (Steinberger and Ježek, 2012). It finds the similarity between two texts by looking at the number of units, such as words or n-grams, that they have in common compared to the number of non-overlapping units they contain. In these evaluations, words are used as the units of overlap.

(4.2) $unit\ overlap(X, Y) = \frac{\|X \cap Y\|}{\|X\| + \|Y\| - \|X \cap Y\|}$

X and Y are the words in the documents being compared. In contrast to ROUGE, unit overlap penalizes an evaluated text for containing words that do not appear in the gold-standard text. A summary will not be rewarded simply for being longer.

The final evaluation measure used is cosine similarity (Steinberger and Ježek, 2012). It is a measure of similarity between documents using vectors of word frequency. Each document is represented by a vector in which each value in the vector is the frequency of a different word in the document. Comparing these vectors gives a measure of how similar two documents are in terms of the words they contain and how frequent those words are.

To evaluate the summaries, each system-produced summary, s, is evaluated against each summary, g, in the set of two corresponding gold-standard summaries. Scores are calculated for each document, and the scores from all documents in the dataset are averaged to produce an overall value, $Average\ M$, for each measure, M.

(4.3) $Average\ M = \frac{\sum_{s \in SystemSummaries} \sum_{g \in GoldStandard} M(s,g)}{2\|SystemSummaries\|}$

In the case of ROUGE, the value is calculated separately between the produced summary and each of the two corresponding reference summaries. Both of these values are included in the calculation of the overall average.

Although not used in this work, another evaluation measure proposed in the literature is the Pyramid Method (Nenkova et al. 2007). This method involves creating summary content units based on multiple human-written summaries. These units are determined by finding information that is repeated in multiple summaries, and the units are weighted based on how many summaries they appear in so that units that occur in more summaries are weighted more highly. This weighting separates information units based on importance, and summaries are scored based on the importance of the content they contain.

2.4 Summarization Process

In order to test how the use of topic structure affects summarization, I explored the impact of topics on the performance of previously proposed algorithms for extractive summarization that are implemented in the Sumy Python library[3]. Specifically, several previously proposed summarizers including LexRank (Erkan and Radev 2004), TextRank (Mihalcea and Tarau 2004), and SumBasic (Nenkova and Vanderwende 2005) were used. These methods will be described in detail below.

As the first step in the summarization process, the complete text was summarized by one of these summarizers. Texts were then divided into topics according to the topics in the RST annotation of the texts. Then each topic was summarized, and the outputs were combined to create a summary of the whole text. These processes are shown in the

[3] https://github.com/miso-belica/sumy

figures below. The first figure shows how texts are summarized when no topics are used, and the second figure shows how topic information is incorporated.

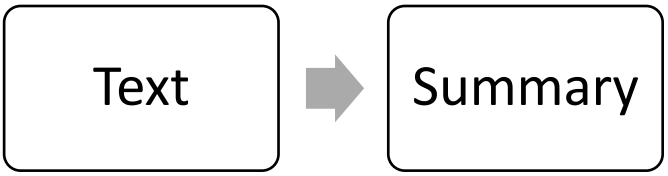


Figure 4.2: Process for summarizing without topics

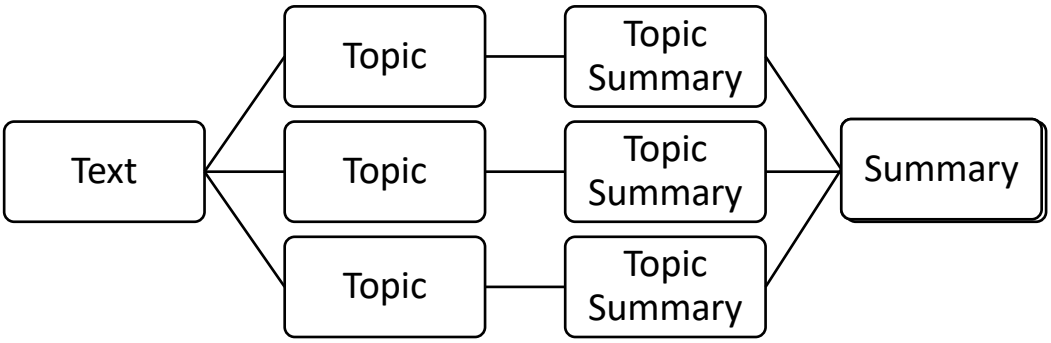


Figure 4.3: Process for summarizing with topics

2.5 Summarization Percentage

The summarizers are given the summary length in terms of the percentage of the original text that should be returned. If the value for the summarization was 20%, the summarizer would return 20% of the original text, where length is measured in sentences. For example, for a text containing 10 sentences, the summarizer would return 2 as the

summary. The same value was used when summarizing the entire text or when summarizing an individual topic. Ideally, this results in similar length summaries whether or not topics are used, because taking 20% of several smaller sections and combining them should be the same as taking 20% of the entire text. One potential problem is with particularly short topics. If a topic contains only a single sentence, the summarizer cannot return 20% of that topic, but will return the sentence, which is 100% of the topic. If there are many of these small topics, the combined result will be a summary that is longer than 20% of the original text.

In more detail, when summarizing any section of text, the summarizer will return either one sentence or the number of sentences given by multiplying the specified percentage by the length of the section, whichever is greater. For example, if the summarization percentage is 10% and a topic contains four sentences, the number of sentences to select is $4 \times 10\% = 0.4$. The maximum of 1 and 0.4 is 1, so 1 sentence will be returned for the summary. This strategy avoids the problem of returning a fraction of a sentence, which is not possible unless sentence reduction or abstractive methods are used. The other possibility in these cases of small sections of text would be to return zero sentences from a section. However, the goal in this work is to emphasize coverage of all topics in a summary. Therefore, the strategy of including at least one sentence from every topic is used. Relatedly, in other cases when the calculation returns a number with a fractional component, such as 2.6, the numbers are rounded down, so that 2 sentences would be chosen. This choice helps prevent summaries from becoming too long. Possible issues of length differences will be discussed in later sections.

The effects of using different values for the summarization percentage were explored. It is interesting to understand how performance changes as the length of the summary changes. Determining the ideal length for a summary is a difficult task as it could differ from one text to another and human judges may not agree on the ideal length. This issue was discussed in more detail in Chapters 1 and 2. Given that there is not a single target summary length for all texts, four values of the summarization percentage were considered: 10%, 20%, 30%, and 40%. Lower values involve choosing less of the text, which means a summarizer must be precise and correct in its choices in order to be similar to the reference summary. At the other extreme, a summary that contains 40% of the original text has more chances to contain the same information as the reference summary, but the inclusion of more information could also result in a summary that contains additional information that should not be in the summary and does not appear in the reference summary. These different percentages highlight the tradeoff between precision and recall. It is important both to contain the information from the reference summary and not to contain extraneous information. Exploring the performance when using different summarization percentages will allow for interesting comparisons of how precise the summarizers are.

How the use of topics interacts with summary length is another interesting question. In a shorter summary, there is less space to convey information, and therefore more pressure to choose the right information. Longer summaries contain a greater total of information, meaning the choice of whether to include any individual sentence is less important. By simply including more sentences, longer summaries are more likely to contain important information from the original text. For example, if a text contains one

crucial sentence that can be judged as the most important sentence to include in a summary, a one-sentence summary has a single chance to make the right selection. On the other hand, a five-sentence summary has five chances. Therefore, precision in choosing the best sentences is more important for shorter summaries than longer summaries. Comparing summaries of different lengths provides another way to understand how effective topics are for summarization. Topics should be more useful when the percentage is lower and summaries are shorter because in those cases it is more important to be precise in the choice of sentences and choose the best sentences that convey the information from the original text and align with the reference summary. As summaries get longer, the standards for which sentences to include in the summary get lower because the summary is less restricted. In that case, additional information such as topic structure may be less useful.

2.6 Summarizers

LexRank (Erkan and Radev 2004) uses a graph-based method for determining the importance of units in a text. The idea behind LexRank is that sentences in a text are related to each other in a type of network where some sentences are more related to each other than others. Sentences that are more similar to the other sentences in the text are considered more central and more important. Cosine similarity is used to judge similarity between sentences. The centrality of a sentence is based on how many sentences it is connected to, by having high cosine similarity values, and how central those sentences are. Therefore, the system takes an input text, produces a similarity-based graph, and selects the most central sentences until the summary length is reached. Erkan and Radev consider several ways of determining centrality. The first is degree centrality, which

looks at how many nodes (sentences) are connected to the node (sentence) under consideration. Another proposed method is LexRank centrality, which builds on degree centrality but also takes into account the centrality of the adjacent nodes so that not all nodes have equal weight in determining centrality. The final method is continuous LexRank, which uses cosine similarity in the construction of the graph so that the graph is weighted. Centrality is then calculated by multiplying the LexRank values by the weights. When evaluated on the data for the Document Understanding Conference (DUC) 2003 and 2004 tasks, LexRank performed comparably to the top systems.

TextRank (Mihalcea and Tarau 2004) is a graph-based ranking algorithm, similar to LexRank, which has been used for many applications. A graph is constructed that represents similarity between sentences. While LexRank uses cosine similarity to measure sentence relatedness, TextRank uses a simple measure of word overlap: the number of words that appear in both of the sentences being compared, divided by the length of the sentences (in order to avoid longer sentences having an advantage). Edges in the graph are weighted according to these overlap scores. The sentences are then ranked according to their centrality, and the highest-ranking sentences are chosen for the summary. When evaluated on the task of single document summarization, TextRank performed better than all but 2 of the 15 systems that participated in the Document Understanding Conference (DUC) 2002.

LexRank and TextRank are similar systems, with their main difference being how similarity is calculated. Some advantages of these graph-based methods are that they are unsupervised and do not require any training data, relying only on information from the text being summarized. They also do not require specific linguistic annotations, which are

not always available. In past work, these methods have been shown to achieve scores in a similar range, with TextRank performing somewhat better than LexRank (Mittal et al.; Kaynar et al. 2017)

SumBasic (Nenkova and Vanderwende 2005) is a summarization system that uses word frequency information. Nenkova and Vanderwende studied how word frequency and content frequency affect how humans choose information for a summary. They found that the words that appear most frequently in the original text typically also appeared in the human summaries, and human summaries contained more of the high-frequency words than state-of-the-art automatic summaries. Based on this importance of frequency for human summaries, SumBasic is a summarization system that uses only frequency information. SumBasic involves several steps. The first is to determine the probability distribution of words in the text, where the probability of a word is estimated as the number of times it occurred in the text divided by the total word count. Sentences are then scored according to the average probability of the words they contain. At this point, the sentence with the highest score that also contains the highest probability word is chosen for the summary. The probabilities of the words in the chosen sentence are then updated to make those words less probable, since they are already included in the summary. The method for updating is shown in the following equation.

(4.4) $p_{new}(word) = p_{old}(word) \cdot p_{old}(word)$

This new probability is considered an approximation of the probability that the word will occur twice in the summary. The process of scoring the sentences, choosing the best sentence, and updating the probabilities is repeated until the specified summary length is reached. This method ensures that the highest frequency words appear in the summary.

Updating the word probabilities ensures that the sentences chosen for a summary at any point depend on which words and information are already contained in the summary, which reduces redundancy. Summaries produced by SumBasic contained even more high-frequency words than human-written summaries. SumBasic has been found to perform well for a relatively simple baseline summarizer using only frequency information. Compared to the other summarizers, there have been mixed results reported in past research with SumBasic performing better than LexRank in some cases (Nenkova and Vanderwende 2005) and worse in others (Griggs 2015).

3 Results and Discussion

3.1 RST Topics vs. No Topics

The following table shows the results of using each of the three different summarizers to summarize texts with and without topics. Each pair of columns shows the result of a different summarizer. The first column in each pair shows the results without using topics, and the second shows the results of using RST topics. The table is also organized by the summarization percentage used, ranging from 10% to 40%. The highest value for each evaluation measure is in bold. Instances in which topics did not improve performance are shaded in gray. Looking at the results shows several interesting effects. Overall, for each measure, in almost all cases the highest value is achieved when using topics. While different summarizers perform slightly better on different measures, it is interesting to note that regardless of evaluation measure or summarizer, the inclusion of topics improves performance.

	LR	LR-T	TR	TR-T	SB	SB-T
10% summarization						
Avg Rouge-1	0.311	0.457	0.413	**0.477**	0.255	0.336
Avg Rouge-2	0.169	0.322	0.288	**0.326**	0.108	0.189
Avg Unit Overlap	0.218	**0.302**	0.271	0.282	0.200	0.233
Avg Cosine Similarity	0.569	**0.660**	0.648	**0.660**	0.529	0.580
20% summarization						
Avg Rouge-1	0.496	0.588	0.554	**0.607**	0.420	0.463
Avg Rouge-2	0.330	0.442	0.415	**0.458**	0.214	0.275
Avg Unit Overlap	0.261	**0.317**	0.260	0.289	0.241	0.260
Avg Cosine Similarity	0.668	**0.711**	0.694	0.710	0.619	0.650
30% summarization						
Avg Rouge-1	0.624	**0.712**	0.682	0.698	0.543	0.556
Avg Rouge-2	0.468	**0.576**	0.563	0.569	0.314	0.347
Avg Unit Overlap	0.265	**0.313**	0.260	0.273	0.244	0.252
Avg Cosine Similarity	0.706	**0.744**	0.722	0.730	0.659	0.677
40% summarization						
Avg Rouge-1	0.708	**0.780**	**0.780**	0.774	0.645	0.649
Avg Rouge-2	0.566	0.651	**0.686**	0.658	0.419	0.442
Avg Unit Overlap	0.255	**0.284**	0.256	0.261	0.238	0.244
Avg Cosine Similarity	0.729	**0.746**	0.743	0.742	0.687	0.697

Table 4.1: Results of using the summarizers with and without topics. LR: LexRank, LR-T: LexRank with Topics, TR: TextRank, TR-T: TextRank with Topics, SB: SumBasic, SB-T: SumBasic with Topics. Highest values for each measure are in bold. Gray cells show when topics did not improve performance

Looking at the results by summarizer, for LexRank, topics always perform better than no topics. Improvements in ROUGE-1 range from 7-15%, ROUGE-2 from 9-15%, unit overlap from 3-8%, and cosine similarity from 2-9%. For TextRank, topics perform better than no topics for all cases, except when the summarization percentage is 40%. For 10% to 30%, improvements in ROUGE-1 range from 2-6%, ROUGE-2 from 1-4%, unit overlap from 1-3%, and cosine similarity from 1-2%. For SumBasic, topics perform better than no topics in all cases. Improvements in ROUGE-1 range from 1-8%, ROUGE-2 from 2-8%, unit overlap from 1-3%, and cosine similarity from 1-5%.

In general, for all summarizers, values of the evaluation measures increase as the percentage increases. However, the increases depend on which measure is considered. ROUGE values increase by the largest margin and with the most consistency. On the other hand, there are smaller increases for the other measures, and they do not always increase. They slightly increase, remain stable, or even slightly decrease as the summarization percentage increases. This effect can be seen in the following plots, which illustrate the results in the table. The first set of three plots shows how ROUGE-1 scores change as the percentage increases, and the second set of three plots shows how unit overlap changes.

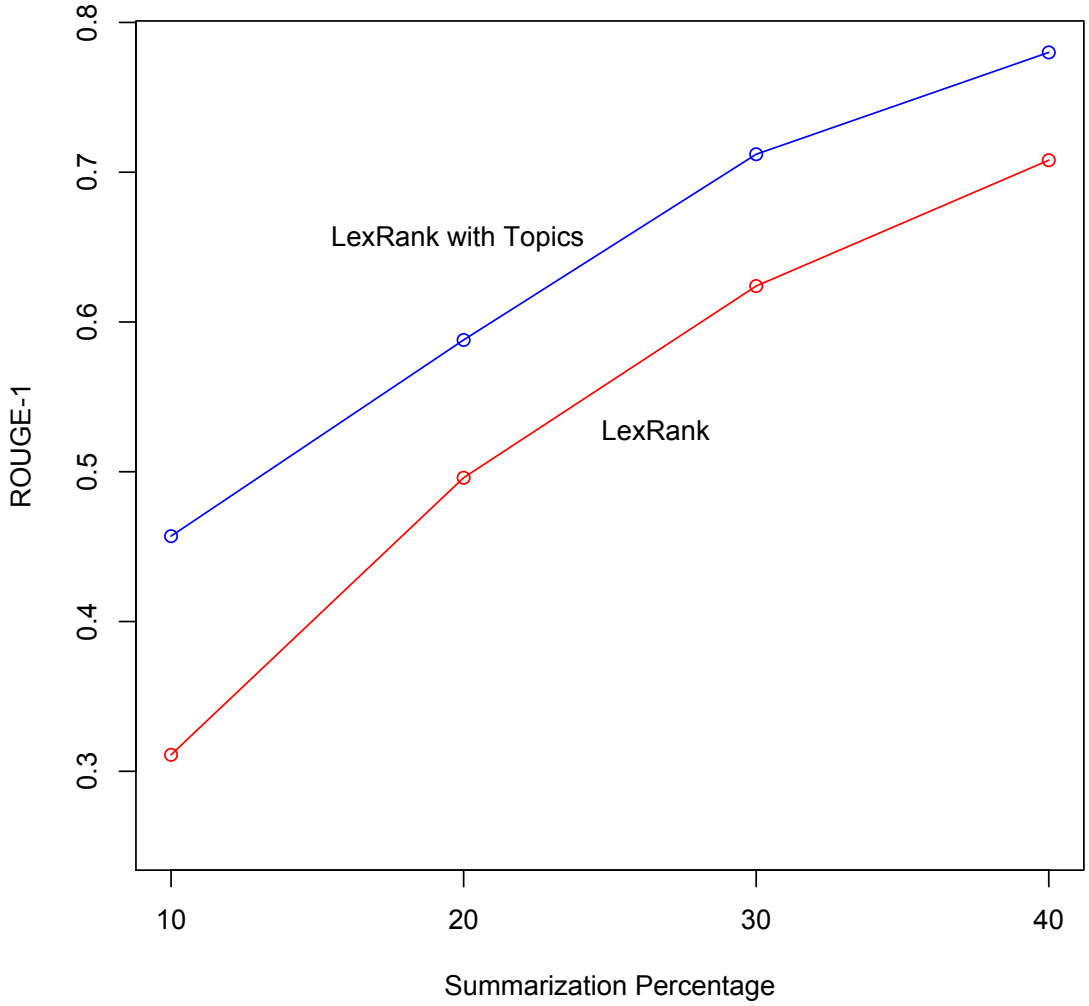


Figure 4.4: ROUGE-1 performance of LexRank as percentage increases

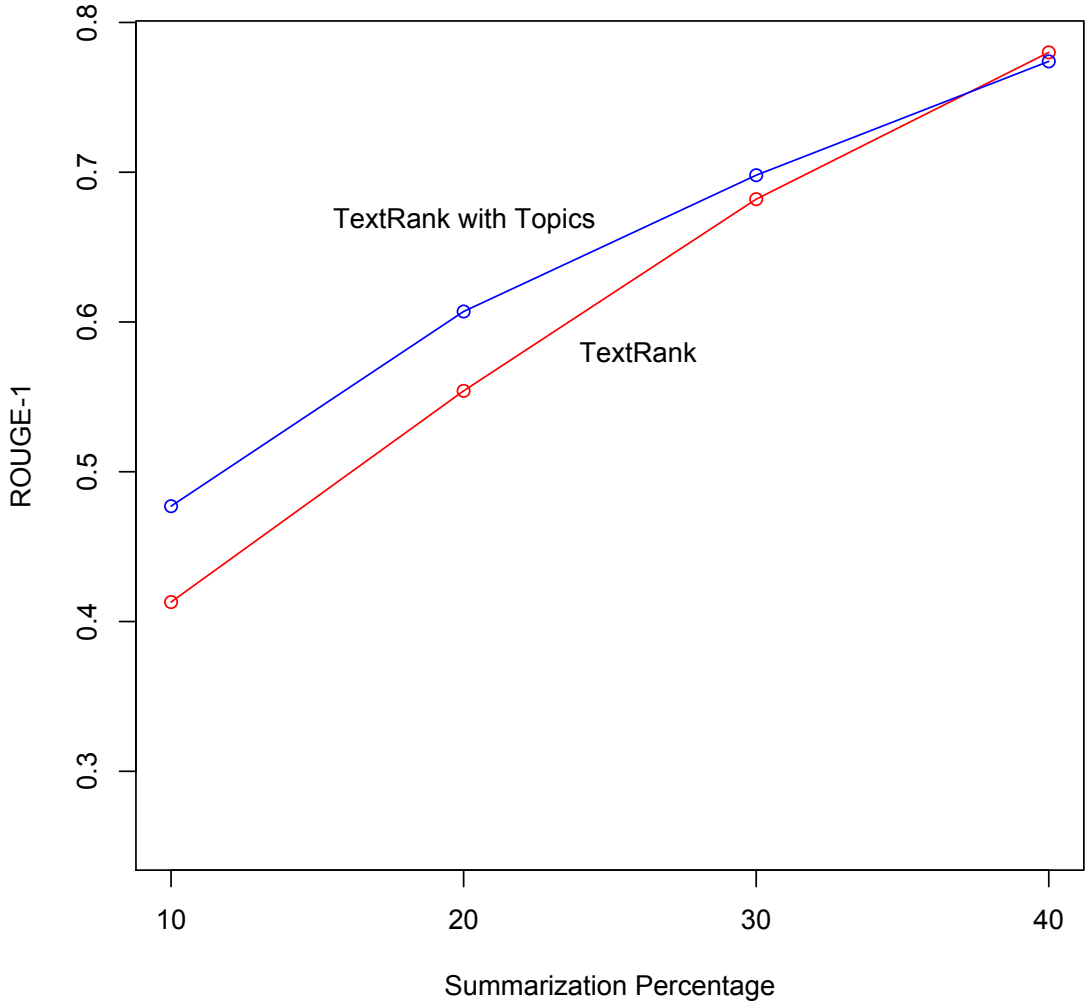


Figure 4.5: ROUGE-1 performance of TextRank as percentage increases

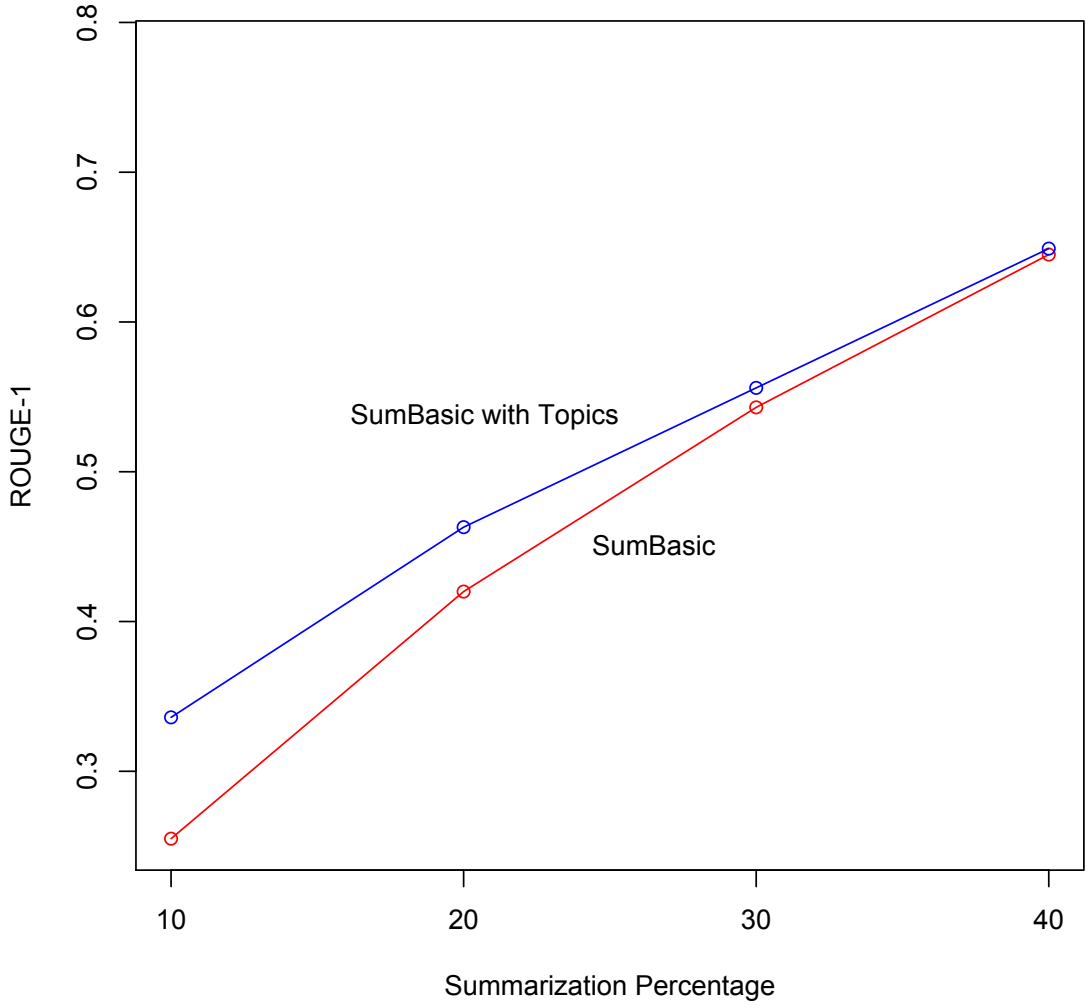


Figure 4.6: ROUGE-1 performance of SumBasic as percentage increases

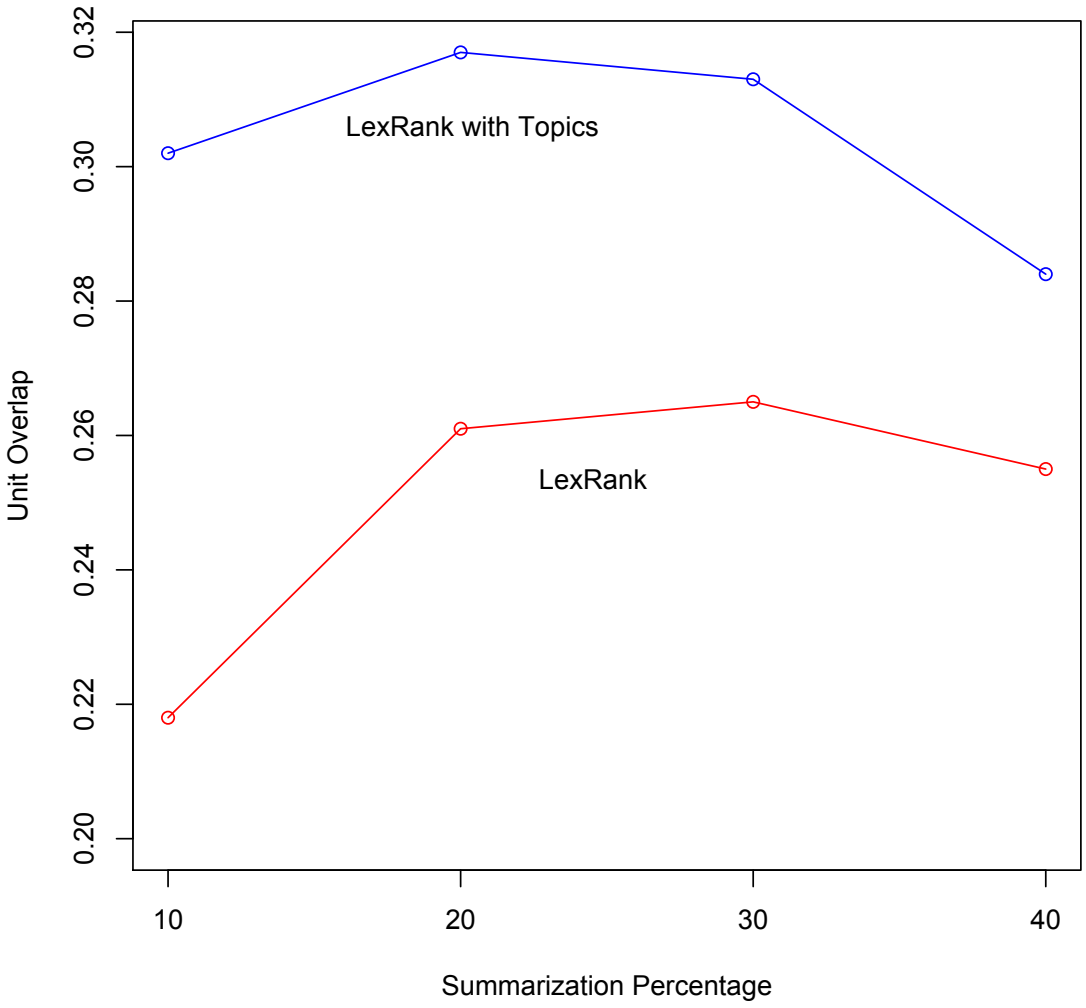


Figure 4.7: Unit overlap performance of LexRank as percentage increases

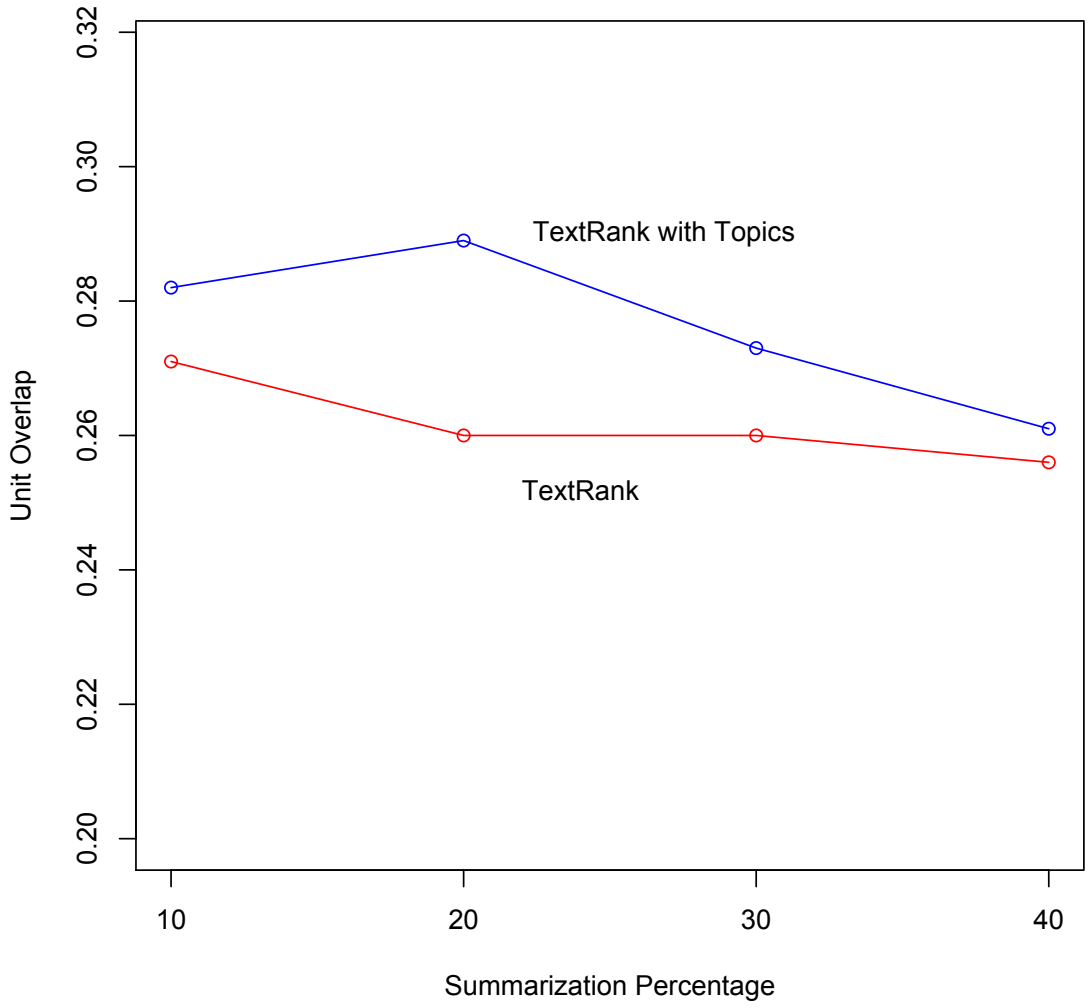


Figure 4.8: Unit overlap performance of TextRank as percentage increases

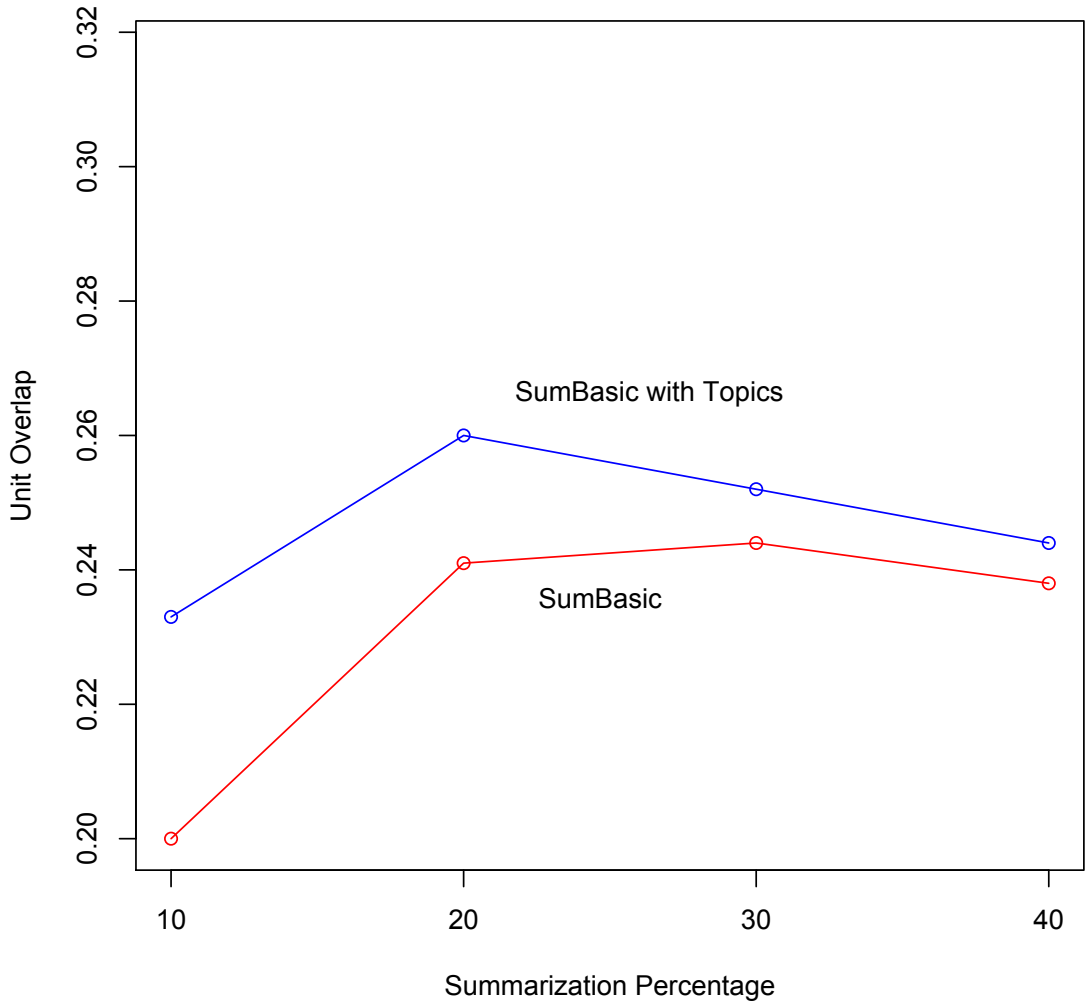


Figure 4.9: Unit overlap performance of SumBasic as percentage increases

ROUGE has such large and consistent increases because it is recall-based, so longer summaries will always perform better. This issue will be discussed further below. The more interesting aspect of the summarization percentages to consider is the difference in the effect of topics for different percentages. The results show that topics create more of an improvement in performance when the percentage is lower and the

summaries are smaller. For all summarizers, the largest improvement in performance is found with 10% or 20% summarization. As the percentage increases, the improvements become much smaller. As discussed above, this suggests that topics do provide useful information for summarization, and that information is the most useful when space is the most limited. At the 10% level, there are improvements of up to 15% by using topics. In contrast, at the 40% level, there are very small improvements or in the case of TextRank, no improvement. Although the improvements are small, it is an interesting finding that for two of the summarizers there is improvement on all measures at the 40% level. Even when this relatively large percentage of the original text is chosen for the summary, the choice of sentences is still improved by using topic information.

The following example shows summarization performed with and without topics. LexRank was used as the summarizer, and the summarization percentage was 30%.

> **Summary without Topics:** But by 1992, he added, Nissan will build one million vehicles a year outside Japan, or sufficient to equal exports. "By the end of the 1990s," he said, "we want to be producing roughly two vehicles overseas for every vehicle that we export from Japan."
>
> **Summary with RST Topics:** Nissan Motor Co. expects net income to reach 120 billion yen (U.S. $857 million) in its current fiscal year, up from 114.6 billion yen in the previous year, Yutaka Kume, president, said. During the next decade, Mr. Kume said, Nissan plans to boost overseas vehicle production sufficiently to account for a majority of sales outside Japan.

The following is the entire text, separated into topics and with EDUs marked for whether they were chosen by the human analysts for a summary (* for one analyst and # for the other).

> **Topic 1:**
> *# Nissan Motor Co. expects net income to reach 120 billion yen
> (U.S. $857 million)
> *# in its current fiscal year, up from 114.6 billion yen in the previous year,

* Yutaka Kume, president, said.
Mr. Kume made the earnings projection for fiscal 1990,
ending next March 31,
in an interview with U.S. automotive writers
attending the Tokyo Motor Show.
The executive said
that the anticipated earnings increase is fairly modest
because Nissan is spending heavily
to bolster its dealership network in Japan
and because of currency-exchange fluctuations.

Topic 2:

*# During the next decade,
Mr. Kume said,
*# Nissan plans to boost overseas vehicle production sufficiently
* to account for a majority of sales outside Japan.
Last year,
Mr. Kume said,
Nissan exported slightly over one million vehicles,
and produced 570,000 cars and trucks at its factories in North America, Europe and Australia.
But by 1992,
he added,
Nissan will build one million vehicles a year outside Japan, or sufficient to equal exports.
"By the end of the 1990s,"
he said,
"we want to be producing roughly two vehicles overseas for every vehicle
that we export from Japan."
That will involve a substantial increase in overseas manufacturing capacity,
he acknowledged,
but didn't provide specific details.

Comparing the system-produced summaries to the human summaries, the summary with topics is more similar to the human summaries. Both of its sentences were selected by both analysts. On the other hand, the summary without topics contains one sentence selected by one analyst and one sentence selected by neither analyst. It is interesting to note that without using topics, the entire summary is composed of sentences from one topic. Using topics ensures that sentences are chosen from different parts of a text, which agrees with what humans do when summarizing as both analysts selected sentences from

both topics. Therefore, using topics improved the summary and made it more similar to human summaries.

The results reported above involved comparing each system-produced summary to two manually-created summaries and including both of these scores in the overall average. However, another possibility is to include only the best of these scores. Each summary is compared to both reference summaries, but only the maximum of these scores is included in the total average over all documents. The results calculated in that way with a 20% summarization percentage are presented in the following table.

	LR	LR-T	TR	TR-T	SB	SB-T
ROUGE-1	0.561	0.655	0.610	0.666	0.473	0.513
ROUGE-2	0.414	0.531	0.490	0.536	0.280	0.338
Unit Overlap	0.301	0.371	0.291	0.331	0.276	0.293
Cos Similarity	0.707	0.752	0.734	0.750	0.658	0.685

Table 4.2: Results when taking maximum value from comparison with two reference summaries

Overall, these results are similar to those reported above. The values are higher in this case because only the maximum values are included in the calculation. However, the sizes of the differences between not using topics and using topics are very similar. One interesting consideration related to these results is the question of whether it is better for a summary to be somewhat similar to multiple reference summaries or very similar to a single reference summary. Since there is variation in which information people select for a summary, being similar to multiple summaries could indicate that a system-produced summary contains a range of information from different summaries and would be accepted by many people. On the other hand, being identical to one manual summary would show that an automatic system can produce summaries that are of the same quality as summaries written by people. Summary evaluation is a difficult task, and considering

similarity to a single reference or multiple references are both useful ways of understanding summary quality. The results in this section suggest that the overall trends in performance, including the improvement seen when using topics, remain the same regardless of which of these approaches is taken.

3.2 RST Topics vs. Random Topics

3.2.1 Results with Random Topics

An important factor to consider when comparing the results of performing summarization with and without topics is summary length. It is possible that summarizing at the topic level could result in summaries of different lengths from the summaries produced by summarizing the entire text. Differences in length could affect these evaluation measures, particularly ROUGE, which is recall-based and therefore benefits from including more words by increasing the chances of having more words in common with the gold-standard. As seen in the results above, the evaluation measures that experienced the greatest improvement in performance are the ROUGE measures, which improved more than unit overlap or cosine similarity. Improvements were found for all measures, but because the greatest improvements were for ROUGE, it is worth investigating whether length is having an effect.

One way of dealing with this potential problem is to compare RST topics with random segmentations of the text. Therefore, in addition to the conditions using RST topics and no topics, a third condition using random topics was tested. Using the topic sizes from the RST topics, texts were randomly divided into topics of the same size. While the RST topics always contain adjacent sentences, the random topics are not constrained in this way. If the topics were contiguous, they could not be both random and

equal in size to the RST topics. This is clear for a text with two topics of equal size. There is only one way to divide topics so that the size condition is met, so the random topics would be identical to the RST topics. Therefore, the random topics are equal in size but do not follow the same adjacency restrictions as the RST topics. This condition provided a control to see whether topic divisions informed by RST information resulted in better summaries than random divisions or whether simply dividing a text into smaller sections with no motivated connection between sentences improves performance. Since the random topics are the same size as the RST topics, length should not have an effect. Summarization was performed at the 20% level because 20% is commonly used as the summarization percentage, and the results above showed strong performance at that level. The following table shows the results of 25 runs with random topics. It includes the values for mean, standard deviation, and the corresponding value when using RST topics. The figure provides a visual illustration of the results. The error bars show two standard deviations below and above the mean. The points represent the values when using RST topics.

LexRank			
	ROUGE-1	Unit Overlap	Cos Similarity
Mean	0.476	0.243	0.659
Standard Deviation	0.010	0.007	0.007
RST Topics	0.588	0.317	0.711
TextRank			
	ROUGE-1	Unit Overlap	Cos Similarity
Mean	0.559	0.266	0.698
Standard Deviation	0.013	0.007	0.006
RST Topics	0.607	0.289	0.710
SumBasic			
	ROUGE-1	Unit Overlap	Cos Similarity
Mean	0.437	0.242	0.641
Standard Deviation	0.014	0.008	0.008
RST Topics	0.463	0.260	0.650

Table 4.3: Values for 25 runs of random topics at summarization percentage of 20%

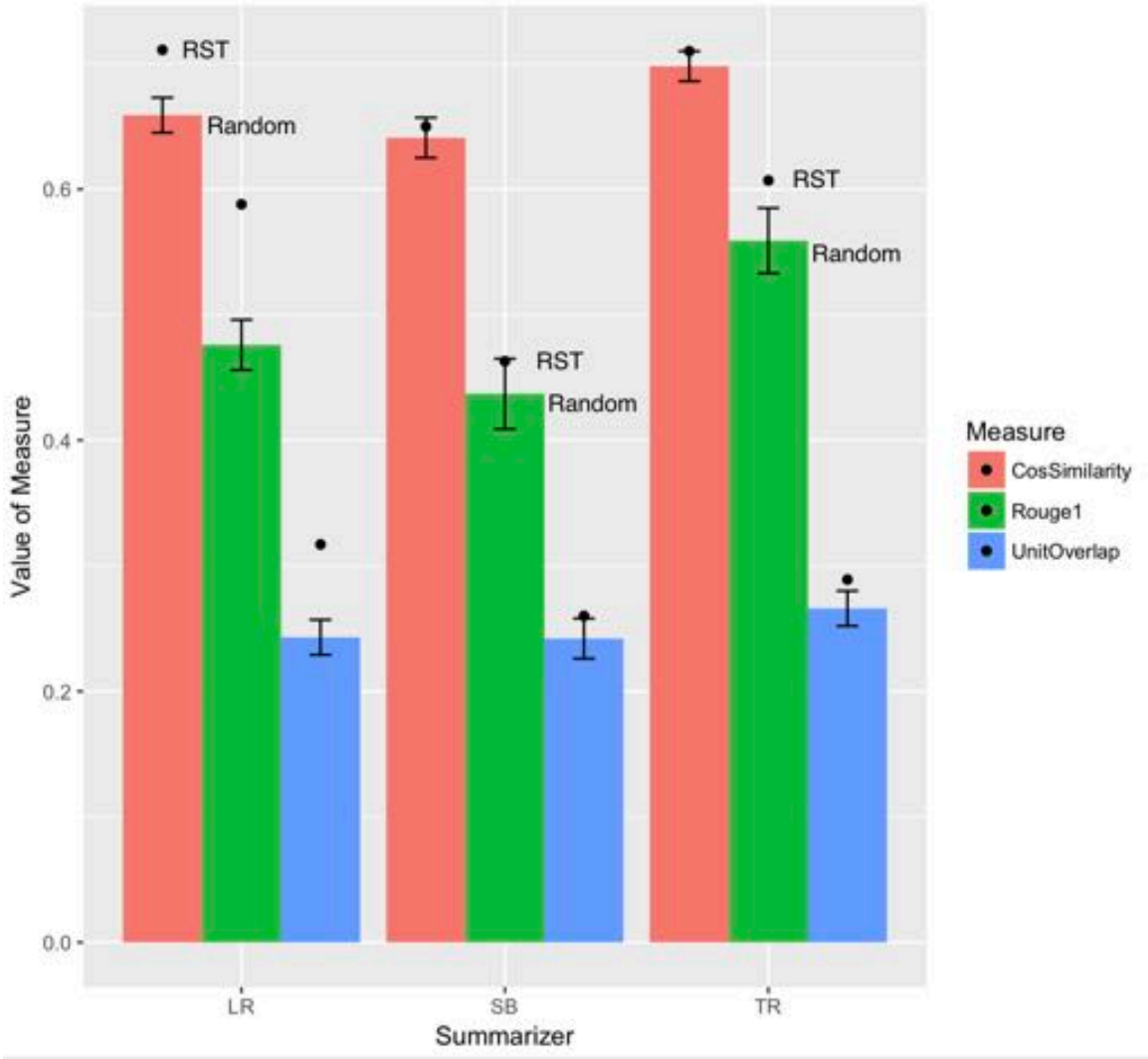


Figure 4.10: Results with random topics. Bars show the mean values, error bars show 2 standard deviations below and above the mean, and points show results with RST topics.

Comparing the mean values to the values with RST topics, the RST values are higher than the random topics for all measures. Looking at the RST values compared to the means + 2 standard deviations, the RST values are greater than or very similar to the random values, indicating that the RST values are significantly different from random. Therefore, the improvement in performance seen with RST topics is not simply due to summarizing smaller sections of the text. RST topics improve performance more than using random topics that are not informed by any structural information.

The following plots provide a visual explanation of the issue. The first plot compares the word count when using RST topics, on the x-axis, to the word count when using random topics, on the y-axis. The second plot compares the word count when using RST topics, on the x-axis, to the word count when no topics are used, on the y-axis. The correlation values are included in each plot. As expected, the RST topics and random topics are more strongly correlated than RST topics and no topics. RST topics tend to produce longer summaries than when no topics are used. This suggests that differences in word count could affect performance when comparing RST topics to no topics as summaries differ more in length. However, looking at random topics provides a control for this problem because the summaries with RST and random topics are more similar in length.

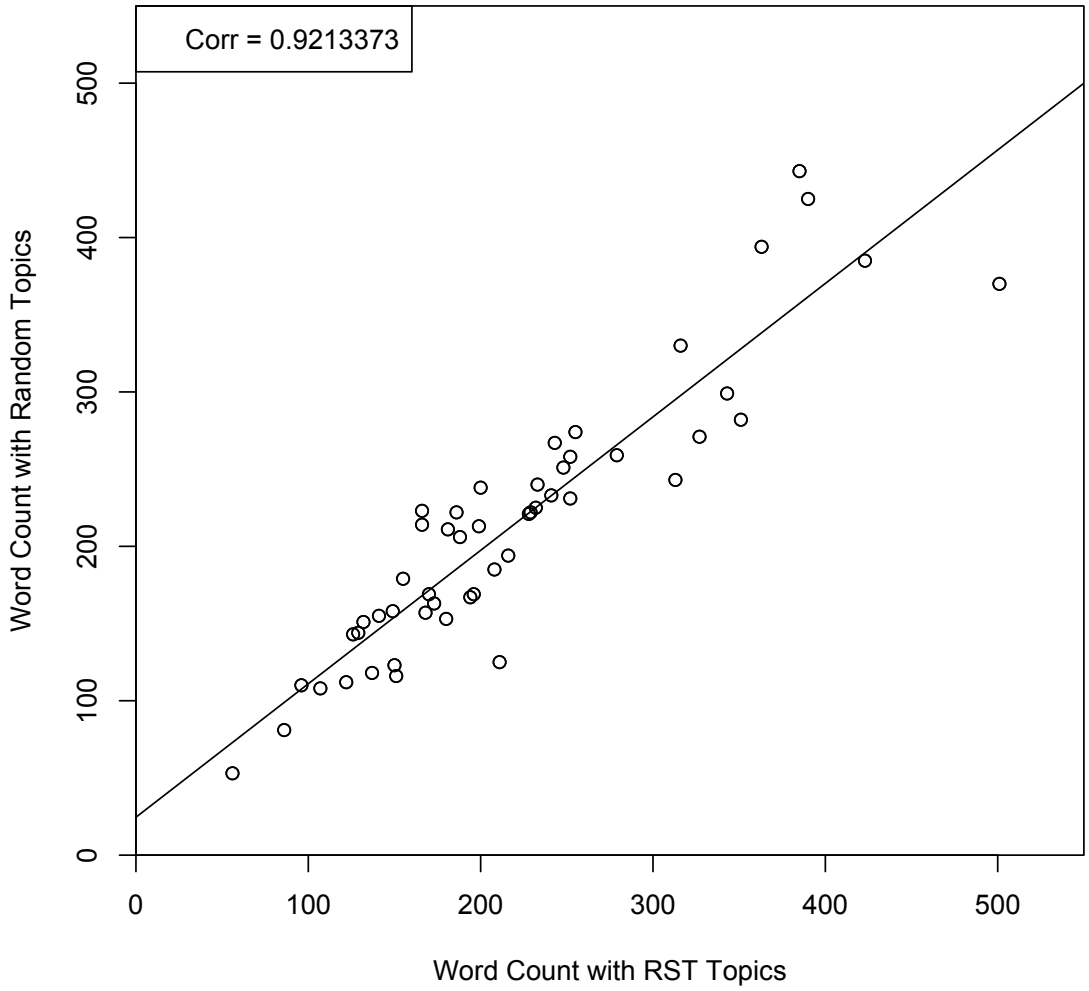


Figure 4.11: Word counts of summaries using RST vs. Random topics

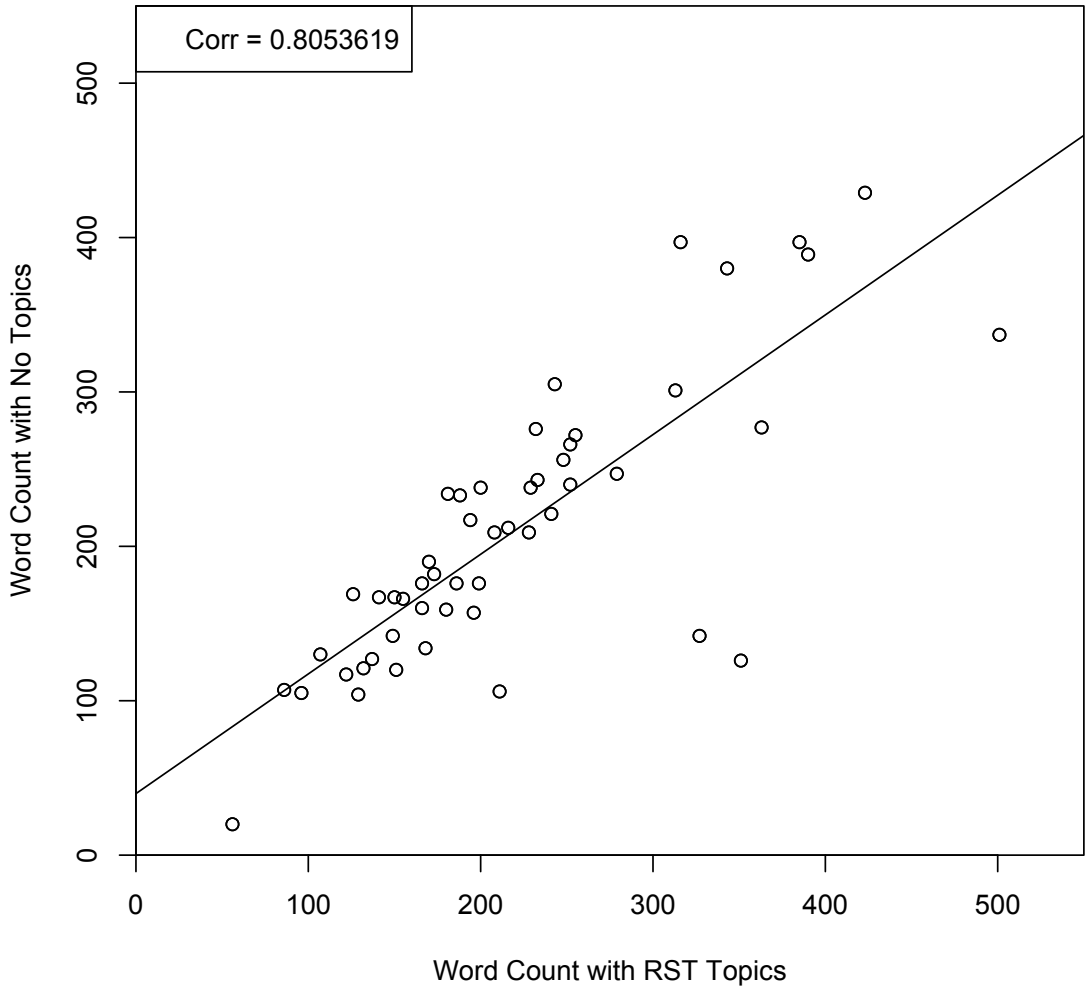


Figure 4.12: Word counts of summaries using RST vs. No Topics

3.2.2 Linear Regression

In addition to comparing RST topics to random topics, another way to explore the effects of length is to consider whether the use of topics has an effect on performance separate from any effect of summary length. A linear regression was performed exploring the effects of different factors on ROUGE and unit overlap to see whether length differences are having an effect. The factors considered were word count in the summary

and RST Topics/Random Topics/No Topics. In the case of random topics, an average of 10 random runs was used. The following tables show the regression results using each summarizer.

LexRank

ROUGE-1		
Factor	Estimate	P-value
Word Count	0.0004995	1.51e-08 ***
Random Topics	-0.01885	0.31
RST Topics	0.08704	4.17e-06 ***
Unit Overlap		
Word Count	-0.0001083	0.0967
Random Topics	-0.01616	0.2508
RST Topics	0.05705	6.32e-05 ***

Table 4.4: Regression with LexRank comparing RST topics, no topics, and average of 10 runs of random topics

TextRank

ROUGE-1		
Factor	Estimate	P-value
Word Count	0.0004331	2.21e-09 ***
Random Topics	0.003793	0.83744
RST Topics	0.05442	0.00345 **
Unit Overlap		
Word Count	-0.00008112	0.0713
Random Topics	0.001569	0.8942
RST Topics	0.02828	0.0170 *

Table 4.5: Regression with TextRank comparing RST topics, no topics, and average of 10 runs of random topics

SumBasic

ROUGE-1		
Factor	Estimate	P-value
Word Count	0.0005227	1.02e-10 ***
Random Topics	0.008811	0.5335
RST Topics	0.03554	0.0121 *
Unit Overlap		
Word Count	-0.00006438	0.2439
Random Topics	0.002969	0.7664
RST Topics	0.01953	0.0505

Table 4.6: Regression with SumBasic comparing RST topics, no topics, and average of 10 runs of random topics

As shown in the tables, word count and RST topics had a significant effect on ROUGE, while random topics had no significant effect. The significance of RST topics shows that using RST topics improves performance compared to not using topics as well as compared to using random topics. As discussed above, ROUGE is affected by differences in word count, and these results show that word count was a significant factor in predicting ROUGE scores. Therefore, differences in summary length seem to be affecting performance. However, while ROUGE is affected by length and word count, unit overlap is not affected in the same way as unit overlap not only rewards summaries for containing words from the gold-standard summaries but also penalizes summaries for including words that do not appear in the gold-standard summaries. The regression results show that word count is not a significant factor in predicting unit overlap scores. For unit overlap the only significant factor is the use of RST topics. These results confirm that RST topics have a significant effect on summarization performance beyond any effect of word count.

3.3 Summary of Findings

These results demonstrate the positive impact that the use of topics has on summarization performance. Specifically, dividing texts into topics using topic relations from RST results in summaries that are more similar to manually-created gold-standard summaries than summarizing texts without the inclusion of topic structure.

By performing summarization at the level of the entire text and at the level of individual topics, I investigated the influence of topic information on summarization performance. A notion of topic that uses information about a text's rhetorical structure in the form of RST relations was explored. The direct comparison of summarization when using topics versus not using topics showed that topic information improves performance. Improvements were found with several evaluation measures, including ROUGE and unit overlap. Performance also improved regardless of which summarizer was used.

The strong performance of the model when using topics has several interesting implications that highlight the contributions of this work. First, the results demonstrate the usefulness of topic structure. Conceptualizing texts as composed of a number of topics not only improves human processing of texts but also increases the quality of summaries produced by automatic systems. In this work, topics were incorporated in a straightforward way, by summarizing a text's topics and combining them to create a complete summary. The results showed that this simple method for including topic information improves performance compared to not using any topics.

Another important finding of this work is the utility of a notion of topic based on rhetorical information. The topics were based on RST relations that connect pieces of a text when the topic has changed between the sections. Using these relations to signal

boundaries between topics proved to be a reasonable method to automatically separate a text into its component topics, and specifically a method that is useful for finding topics relevant for summarization. The improvements in performance seen with this notion of topic also demonstrate another way that rhetorical information such as RST can be used as part of the summarization process.

4 Topics using LSA

Given the improved summarization performance seen when using topics based on RST, it is worth considering whether other notions of topic, particularly common topic modeling methods, are useful for this task. Another notion of topic that was tested is based on Latent Semantic Analysis (LSA), a topic modeling method described in the previous chapter. LSA provides an interesting comparison because it is easily automated compared to RST topics, which rely on manual annotations. The details of how LSA works are presented in Chapter 3. This section describes how LSA was used to divide texts into topics.

4.1 Dividing into Topics using LSA

An implementation of LSA from Gensim topic modeling software was used (Rehurek and Sojka 2010). The training corpus used was the set of Wall Street Journal articles from the Penn Treebank (Marcus et al. 1999). This data was chosen because the test data from the RST Treebank also consists of WSJ articles, and therefore the model will be trained on data with similar style and content. This training data consisted of 2451 articles. Several pre-processing steps were performed before running LSA on these texts. First, stopwords were removed, as well as words that occurred only once in the dataset. Stopwords are common words, typically function words, that are likely to occur in many

documents but do not provide useful information for characterizing the content of texts or differentiating between texts. Words that occur only once in the corpus are also less likely to be useful for discovering the types of semantic relations found by LSA because they only occur in one context, and LSA relies on context for word meaning. After removing these words, the training data consisted of 21,888 unique words.

Another type of processing was also performed. The texts are first represented as vectors of word frequencies. These frequency values are changed using Term Frequency Inverse Document Frequency (TFIDF), a method discussed previously which will be reviewed here. This is a technique that weights words based on how frequently they occur in a document but also takes into account how frequently those words appear in the entire corpus of documents. If a word occurs many times in a document it would generally be judged as important for the document and useful for describing the document. However, if that word occurs many times in all of the documents in the corpus, it is not useful for describing any particular document relative to the rest of the corpus. TFIDF makes such words less important by giving them less weight. At the same time it gives more weight to words that may have occurred less frequently in a document but perhaps did not occur in any other document in the corpus, and are therefore better candidates for characterizing this particular document.

In mathematical terms, TFIDF is the product of two values, term frequency and inverse document frequency. Term frequency is simply the frequency count of how many times a word, t, occurred in a document, d.

(4.5) $term\ frequency(t, d) = \ count(t, d)$

Inverse document frequency is calculated by taking the log of the total number of documents, $|D|$, divided by the number of documents containing word t, $|T|$. IDF will be lower for more words that occur in more documents and higher for words that occur in fewer documents.

(4.6) $inverse\ document\ frequency(t, D) = log \frac{|D|}{|T|}$

These two values are multiplied to produce TFIDF.

(4.7) $TFIDF(t, d, D) = term\ frequency(t, d) * IDF(t, D)$

TFIDF is used to transform the frequency vectors before performing LSA to take into account how frequently words occurred in particular documents as well as the entire corpus.

An important part of LSA is the number of dimensions that are used when reducing the semantic space. This number can also be thought of as the number of topics that the method will find in the data. Different numbers of dimensions have been suggested in the literature, ranging from 10-1000 (Landauer and Dumais 1997; Bradford 2008). Given the size of the training data and common values suggested in previous research, three values for the number of topics were tested: 50, 100, and 200.

An important aspect of testing LSA as a method for determining topics for summarization is to compare it with the proposed RST method. To enable a direct comparison between the two methods, the number of topics found for each document in the test set (the same set of texts used with the RST topics) was kept the same as the number of topics in the RST annotation of each document. This was implemented in two different ways. Although RST topics only contain adjacent sentences, the LSA topics were not constrained to include only contiguous sentences.

In the first method, the LSA model was trained, and a number of dimensions was specified, 100 for example. Then the correct number of topics, n, for each document was taken from the RST annotation. The Gensim implementation of LSA produces values for how related each document is to each of the dimensions in the model, so it is possible to determine which of the dimensions were most relevant for a particular document. Using that information, the n dimensions out of the total dimensions that were most related to a document were selected as that document's topics. For example, if a document had two topics in the RST annotation, the two dimensions with the highest relevance values for that document were selected as its topics. Then, going within a document to the sentence level, a similar process was performed to find which of these topics each sentence was most related to. Sentences were grouped into topics according to which of the document's topics they were most similar to. Overall, the most relevant topics for a document were selected from the full set of topics (dimensions) for the corpus, and then sentences were divided into topics based on their relatedness to this smaller set of topics. The idea of this method is that each document contains a small set of topics out of all the topics present in the dataset, so these topics are first found and then the sentences of the document are grouped with these topics.

The other method skips the step of finding the most relevant topics for a document and works directly at the sentence level. As in the first method, the number of topics, n, to choose for each document was taken from the RST annotation. Then this number was used as k in a k-means clustering algorithm. Clustering was performed over the sentence vectors that represent a document's sentences in the LSA semantic space. These vectors contain values for how related a document is to each of the topics or

dimensions in the model. Clustering divides the sentences of a text into *k* topics based on similarity of the sentence vectors. Clustering was performed using scikit-learn (Pedgregosa et al. 2011).

4.2 Results with LSA Topics

Once texts were divided into topics, each topic was summarized using one of the summarizers: LexRank, TextRank, or SumBasic. Then the topic summaries were combined to create a summary of the whole text. The following tables present the results. For each summarizer and number of LSA dimensions (50, 100, or 200), the model was run ten times, and the mean and standard deviation of each evaluation measure were calculated. The results reported above when no topics were used and when RST topics were used are also repeated at the bottom of each table for reference. To visualize the results, graphs showing the ROUGE-1 and unit overlap scores are included for each method using LexRank.

Using Method 1:

LexRank:

	ROUGE-1	ROUGE-2	Unit Overlap	Cos Sim
50 Dimensions				
Mean	0.496	0.321	0.259	0.677
Std Dev	0.006	0.010	0.004	0.004
100 Dimensions				
Mean	0.487	0.311	0.255	0.672
Std Dev	0.009	0.011	0.005	0.005
200 Dimensions				
Mean	0.487	0.311	0.256	0.670
Std Dev	0.009	0.013	0.006	0.006
No Topics	0.496	0.330	0.261	0.668
RST Topics	0.588	0.442	0.317	0.711

Table 4.7: LSA results using LexRank, using first method of dividing into topics

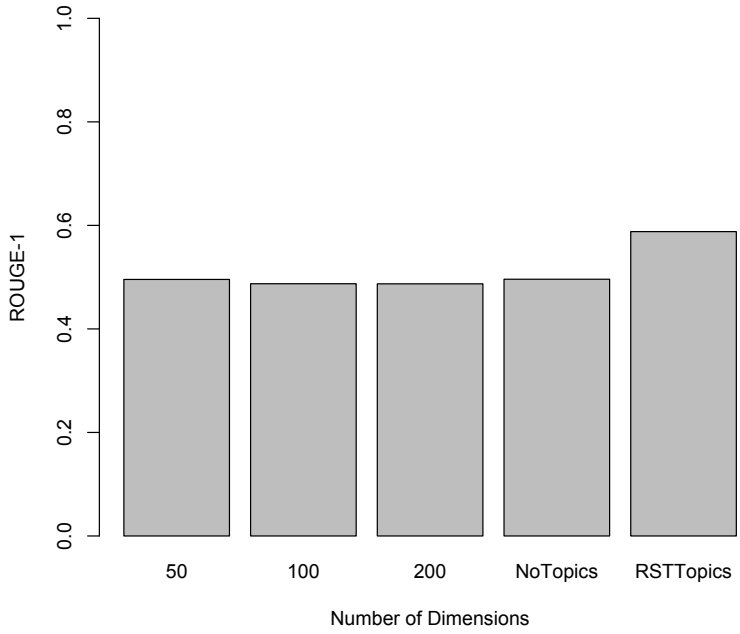


Figure 4.13: ROUGE-1 results using LexRank

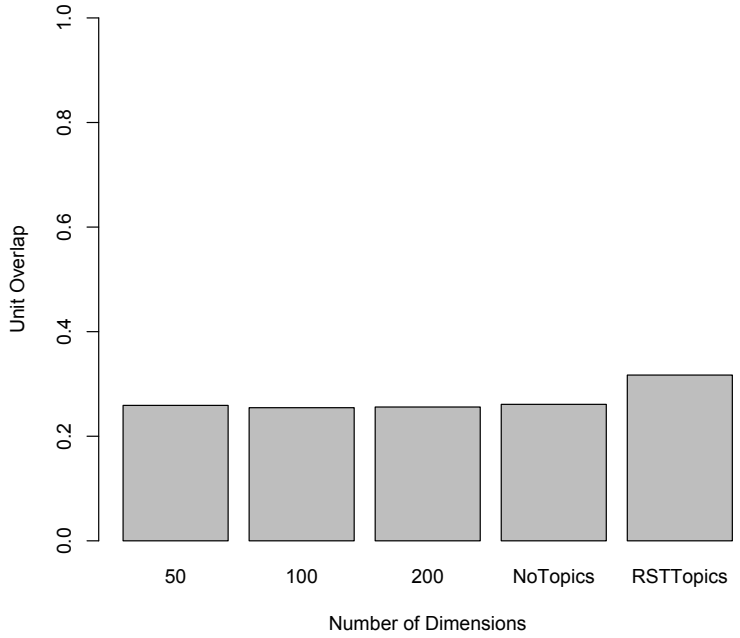


Figure 4.14: Unit overlap results using LexRank

TextRank:

	ROUGE-1	ROUGE-2	Unit Overlap	Cos Sim
50 Dimensions				
Mean	0.553	0.399	0.266	0.698
Std Dev	0.009	0.012	0.004	0.005
100 Dimensions				
Mean	0.554	0.399	0.267	0.699
Std Dev	0.008	0.011	0.004	0.004
200 Dimensions				
Mean	0.553	0.397	0.268	0.702
Std Dev	0.006	0.008	0.004	0.003
No Topics	0.554	0.415	0.260	0.722
RST Topics	0.607	0.458	0.289	0.710

Table 4.8: LSA results using TextRank, using first method of dividing into topics

SumBasic:

	ROUGE-1	ROUGE-2	Unit Overlap	Cos Sim
50 Dimensions				
Mean	0.431	0.233	0.243	0.629
Std Dev	0.009	0.012	0.004	0.005
100 Dimensions				
Mean	0.439	0.244	0.248	0.637
Std Dev	0.007	0.010	0.004	0.005
200 Dimensions				
Mean	0.435	0.238	0.245	0.635
Std Dev	0.005	0.005	0.002	0.003
No Topics	0.420	0.214	0.241	0.619
RST Topics	0.463	0.275	0.260	0.650

Table 4.9: LSA results using SumBasic, using first method of dividing into topics

Method 2:

LexRank:

	ROUGE-1	ROUGE-2	Unit Overlap	Cos Sim
50 Dimensions				
Mean	0.516	0.351	0.261	0.674
Std Dev	0.007	0.009	0.004	0.003
100 Dimensions				
Mean	0.521	0.354	0.265	0.678
Std Dev	0.012	0.013	0.005	0.005
200 Dimensions				
Mean	0.524	0.357	0.269	0.682
Std Dev	0.008	0.010	0.005	0.003
No Topics	0.496	0.330	0.261	0.668
RST Topics	0.588	0.442	0.317	0.711

Table 4.10: LSA results using LexRank, using second method of dividing into topics

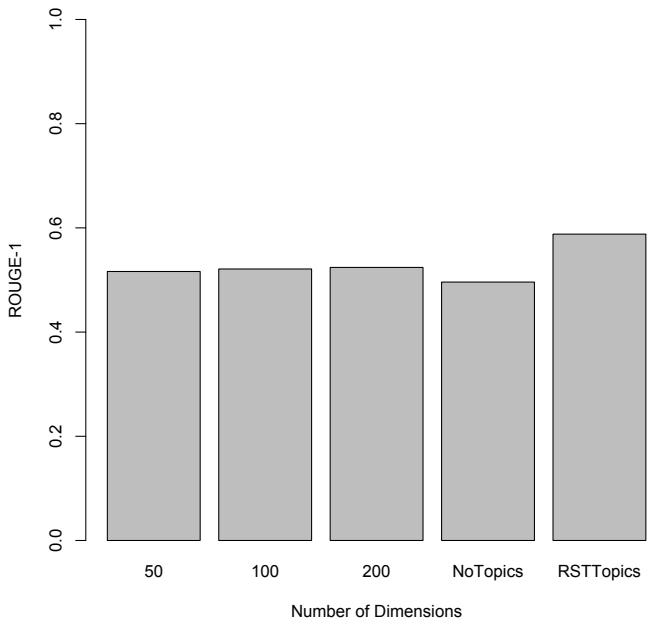


Figure 4.15: ROUGE-1 results using LexRank

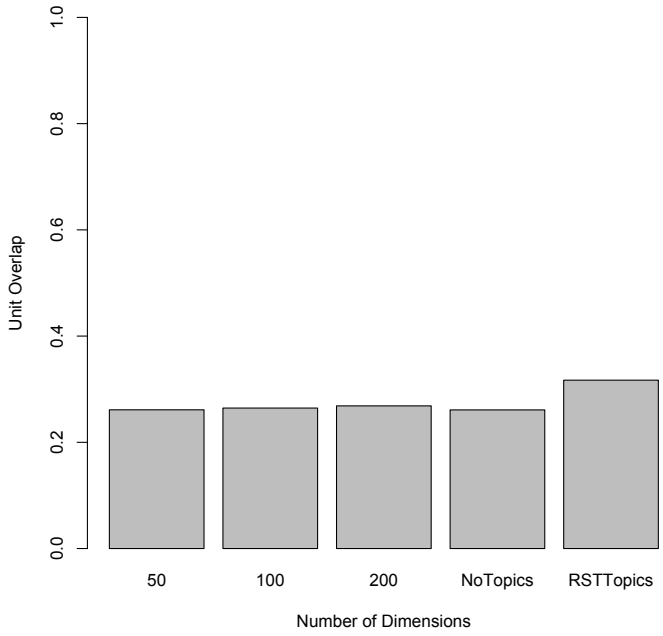


Figure 4.16: Unit overlap results using LexRank

TextRank:

	ROUGE-1	ROUGE-2	Unit Overlap	Cos Sim
50 Dimensions				
Mean	0.554	0.391	0.256	0.695
Std Dev	0.008	0.012	0.004	0.004
100 Dimensions				
Mean	0.556	0.391	0.255	0.693
Std Dev	0.008	0.012	0.004	0.003
200 Dimensions				
Mean	0.551	0.385	0.253	0.691
Std Dev	0.009	0.014	0.005	0.003
No Topics	0.554	0.415	0.260	0.722
RST Topics	0.607	0.458	0.289	0.710

Table 4.11: LSA results using TextRank, using second method of dividing into topics

SumBasic:

	ROUGE-1	ROUGE-2	Unit Overlap	Cos Sim
50 Dimensions				
Mean	0.431	0.237	0.235	0.628
Std Dev	0.006	0.008	0.004	0.004
100 Dimensions				
Mean	0.430	0.231	0.236	0.628
Std Dev	0.007	0.009	0.004	0.005
200 Dimensions				
Mean	0.442	0.246	0.244	0.632
Std Dev	0.009	0.011	0.005	0.005
No Topics	0.420	0.214	0.241	0.619
RST Topics	0.463	0.275	0.260	0.650

Table 4.12: LSA results using SumBasic, using second method of dividing into topics

4.3 Discussion

Looking across all summarizers and numbers of dimensions, RST topics perform better than LSA topics. The difference in performance is evident for all evaluation measures. In general, LSA topics perform similarly to using no topics at all. Since there are a large number of results, this section will highlight a few specific cases as examples. Using the second method for topic division with LexRank as the summarizer and 50 dimensions in the model, the mean ROUGE-1 score is 52%, and the mean unit overlap is 26%. Without using any topics, the corresponding scores are 50% and 26%. With RST topics, the scores are 59% and 32%. Comparing LSA to no topics, LSA performs better on ROUGE-1 and equal on unit overlap. Looking at RST topics, RST performs better on both measures. To consider an example with different settings, with the second method for topic division with TextRank as the summarizer and 200 dimensions in the model, the mean ROUGE-2 score is 39% and the mean unit overlap is 25%. The corresponding values without using topics are 42% for ROUGE-2 and 26% for unit overlap. With RST topics, the values are 46% for ROUGE-2 and 29% for unit overlap. LSA performs worse

than no topics and RST topics on both measures. These examples highlight the types of performance differences found across all summarizers and numbers of dimensions.

There is not much variation between different numbers of dimensions used in the LSA model. For example, for Method 2 with LexRank, ROUGE-1 has a value of 51% for 50, 100, and 200 dimensions, and the value of unit overlap for all dimensions is 26-27%. The similar performance is likely because each document will only be strongly related to a small set of the total topics. Therefore, increasing the total number of topics will not greatly affect the number of topics relevant to a particular document or the strength of those relations. The results are also very similar for both topic division methods used with LSA.

One difference between the LSA topics described above and the RST topics is the fact that the RST topics consist only of adjacent sentences, while the LSA topics are not limited in this way. On the one hand, the fact that LSA topics can include sentences from any part of the text captures an important difference between the LSA notion of topic and the notion of topic from RST. On the other hand, there could be effects on performance due to the differences in adjacency compared to non-adjacency. In addition to the methods described above for using LSA to divide texts into topics, I also explored a method for using LSA to divide into topics in a way that preserves sentence adjacency. Based on the idea of using LSA to determine coherence between sentences (Landauer et al. 1998), boundaries were found between topics by finding the positions with the least coherence. The LSA sentence vectors for all pairs of adjacent sentences were compared, and the sentences with the least similarity were selected as boundaries between topics. Since this method requires specifying the number of intended topics, the number of topics

that were in each text according to its RST annotation was used. That number determines the number of boundaries to place, and dividing at those boundaries results in topics. When testing this method, the results were very similar to the LSA results using the other methods to divide into topics. Using these topics did not result in any significant improvement in summarization performance. Therefore, the distinction between whether topics include adjacent or non-adjacent sentences does not seem to have an impact on whether topics determined through LSA are useful for this task.

In general, the types of topics found by using LSA are not very useful for a summarization system that uses topics. Since RST topics outperform LSA topics, it is interesting to contrast the two notions of topic. While LSA and other topic modeling methods have been successfully used in previous research, it is important to consider the types of tasks that these models have been used for. They have been used for information retrieval tasks such as finding documents that are most related to a query (Deerwester et al. 1990; Dumais 1994; Dumais 1995). They have also been used for document clustering and classification (Bellegarda 2000; Zelikovitz and Hirsh 2001; Song and Park 2007; Wei et al. 2008). All of these tasks focus on finding similarities or differences at the level of the entire text. For example, the goal of a classification task might be to divide a set of documents into those that are related to computer science and those that are related to history. As these are relatively distinct topics, it should be fairly simple using methods such as LSA to decide whether a document is more related to one topic or the other. Zelikovitz and Hirsh (2001) performed this type of classification task by classifying the titles of physics papers by sub-discipline. In these tasks, all documents in the corpus

belong to one of a small set of categories, astrophysics or condensed matter in the physics example.

On the other hand, summarization operates within a single document. A single text can contain multiple topics, a fact which is an important basis for the work in this paper. However, these topics are unlikely to be as distinct as computer science and history. They likely involve more subtle differences such as different perspectives on a single issue or different aspects of an idea, such as a movie review, which discusses the actors, plot, and music of one movie. This difference is mainly a question of granularity. Topics at the level of an entire document are likely to be broader and easier to distinguish from topics of other documents while topics within a document are likely to be narrower and more similar because they all relate to the main topic of the document. It is also not the case that each topic in a text can be classified into one of a small set of known categories as in the physics example. In this case, a method based on word distributions such as LSA might not be sufficient to distinguish between different topics of a single document as all topics are likely to include words related to the main idea of the document. In addition, topics are smaller than documents, meaning there are fewer words to inform decisions. LSA seems to be better suited for tasks in which the texts it is being used to compare are relatively different and belong to one of a set of labeled categories. The results found when using LSA to find topics to use as part of a summarization system suggest distribution-based topic modeling methods are not useful for the smaller scale task of differentiating between topics within a single coherent document.

5 Using an Automatic RST Parser

5.1 Motivation and Parser Details

Given the positive results found when using topics informed by RST for summarization, an important question is how this method can be applied to a broader set of data. In an ideal situation, a summarization system would be able to handle any document as input. However, RST annotation is a work-intensive process. The RST Discourse Treebank was created by trained analysts who determined the structure and relation types of the documents and manually labeled them. For that reason, this dataset is used as the gold standard for research involving RST. However, manually producing RST annotations for a document before summarizing it is unrealistic as that would require time and training. Therefore, the number of documents that can be summarized using RST topics is currently limited.

One way to overcome this issue is to use an automatic RST parser. Several parsers have been proposed and implemented (Feng and Hirst 2012; Feng and Hirst 2014; Hernault et al. 2010; Ji and Eisenstein 2014). For example, the Feng and Hirst (2012) parser works by taking a bottom-up approach to building a discourse tree. A classifier decides whether two adjacent EDUs are likely to be the arguments of a relation and those that are likely related are connected in a subtree. Then a classifier determines how to label the relation. These classifiers continue to be applied until all units in the text have been combined into a discourse tree. The classifiers are trained on the RST Discourse Treebank and use features such as n-grams, part of speech tags, word similarity, and the presence of cue phrases.

If an automatic parser could be used instead of requiring manual annotations, then the topic information given by RST could be used in the automatic summarization of any document. The document would first be parsed, and the output would be given to the summarizer.

The viability of this approach will depend on the performance of the parser. Automating steps in a process can improve the speed with which tasks are performed as well as the amount of data that can be processed. However, automation also introduces noise. Different tools vary in how closely their automated output resembles human performance on the same task. When a tool, such as a parser, does not have perfect performance, it introduces noise into the system that can affect later steps in the process. The more automated steps that a system contains, the more opportunities that exist for noise. When evaluating the final output of a system with multiple steps, it can be difficult to separate the meaningful results from the effects of the noise in the system. The specific concern in this case is the parser's accuracy in identifying topic relations. If the parser struggles to identify these relations, the utility of the parser's output for the topic summarization method proposed in this work will be limited. Therefore, it is important to consider the performance at different points in the process and be aware of how errors at one point will affect the rest of the system. The performance of automatic parsers on topic relations in particular will be discussed below.

In order to test the possibility of automatically producing RST topic information to use for summarization, I used an RST parser from Feng and Hirst (2012) to produce RST annotations. The results of using parser-produced topic divisions were compared to using the gold standard divisions from the RST Discourse Treebank. A useful feature of

the parser implemented by Feng and Hirst is the ability either to provide gold standard elementary discourse units (EDUs) for the parser to use when creating the discourse structure or for the parser to perform its own EDU segmentation and parse the resulting EDUs into an RST structure. Both options were tested for comparison. All experiments followed the same process as above, with texts divided into topics, topics summarized using LexRank, TextRank, or SumBasic, and the output combined to create an overall summary. A summarization percentage of 20% was used.

5.2 Results

The following table shows the results of using the parser with gold standard EDU segmentations. Looking at the documents in the RST Treebank that had topics found by the parser and have extractive reference summaries resulted in a dataset of 33 documents. The experiments were performed on this dataset to allow a direct investigation into whether the topics found by the parser are useful for summarization compared to not using topics.

	LR	LR-FH	TR	TR-FH	SB	SB-FH
Avg ROUGE-1	0.494	0.520	**0.578**	0.541	0.435	0.454
Avg ROUGE-2	0.306	0.351	**0.433**	0.375	0.238	0.271
Avg Unit Overlap	0.254	0.269	**0.275**	0.257	0.249	0.251
Avg Cos Similarity	0.672	0.693	**0.712**	0.696	0.639	0.661

Table 4.13: Results of using automatic parser with gold standard EDU segmentations. LR-FH: LexRank using topics from Feng Hirst parser, TR-FH: TextRank using topics from Feng Hirst parser, SB-FH: SumBasic using topics from Feng Hirst parser.

The results show some improvement when using topics compared to not using topics. However, the improvements are smaller and less consistent than those seen previously with manually-annotated topics. Looking by summarizer, LexRank and SumBasic both have better results when topics are used, while TextRank shows a decrease in performance. The highest values overall are found by using TextRank

without topics. Overall, these results provide some support for using the topic relations found by an RST parser as part of a summarization system. While the improvements are not as large or consistent as when using the gold standard topics, there is evidence of some improvement with parser topics.

After testing the summarization system with the manual EDU segmentation from the RST Treebank, the other option to test involves the additional step of dividing a text into EDUs. The following table shows the results when the RST parser performs the EDU segmentation and uses those EDUs to parse the text into relations. Using the topics found in this way resulted in 27 documents that have topic relations and also have gold standard extractive summaries. The table shows the results of performing summarization on this set of documents, with and without topics.

	LR	LR-FH	TR	TR-FH	SB	SB-FH
Avg ROUGE-1	0.507	0.479	**0.582**	0.524	0.431	0.424
Avg ROUGE-2	0.316	0.305	**0.430**	0.359	0.215	0.226
Avg Unit Overlap	0.253	0.257	**0.263**	0.254	0.239	0.238
Avg Cos Similarity	0.666	0.649	**0.696**	0.681	0.617	0.635

Table 4.14: Results of using automatic parser with automatic EDU segmentations. LR-FH: LexRank using topics from Feng Hirst parser, TR-FH: TextRank using topics from Feng Hirst parser, SB-FH: SumBasic using topics from Feng Hirst parser.

Compared to the results in the previous table, using topics results in decreases in performance in most cases. There are a few small improvements, such as for unit overlap with LexRank and cosine similarity with SumBasic. In general, automatically parsed topics do not improve performance. The set of documents is slightly different from those used to produce the results in the previous table, but comparing these two different versions of the parser suggests that the parser performs better when given gold standard EDU segmentations. An interesting finding is that the EDU segmentations found by the parser are not the same as the segmentations in the RST Treebank. EDU segmentation is

commonly described as an easy problem (Feng and Hirst 2014), but this segmenter clearly produces different output because the parsers differ on the topics they find. These results illustrate the problem discussed at the beginning of this section. As more parts of the process are automated, the performance decreases. With the first option, the parsing into RST relations and structure is automated but the EDU segmentation is not. Performance is somewhat better than not using topics, but the improvements are not as great as when using topics from the Treebank. In the second option, both segmentation and parsing are automated. In that case, performance is worse than when only parsing is automated. In fact, there is almost no improvement in performance compared to not using topics at all.

In order to directly compare the usefulness of the topics found by the parser to the human-annotated topics, I looked at the set of documents that have topic relations in the Treebank as well as topics found by the parser. Restricting the documents to those that have both kinds of topics and also have gold standard summaries results in a very small dataset of only 18 documents. Therefore, this exploration is not meant to produce very strong conclusions but rather to provide a general sense of how these different topics compare. Limiting the documents under consideration in this way allows for a very direct comparison of the parser topics to the Treebank topics since all documents in this set have both kinds of topics. Any differences in performance are due to how the texts are divided into topics rather than simply the presence or absence of topics. The following table shows three settings for each summarizer: no topics (LR, TR, SB), topics found by the parser (LR-F, TR-F, SB-F), and topics from the RST Treebank (LR-T, TR-T, SB-T).

These results are from using the first version of the RST parser in which the gold standard EDU segmentation is provided to the parser.

	LR	LR-F	LR-T	TR	TR-F	TR-T	SB	SB-F	SB-T
R1	0.488	0.510	**0.585**	0.575	0.555	**0.585**	0.425	0.454	0.447
R2	0.313	0.348	0.432	**0.445**	0.400	0.429	0.229	0.277	0.261
UO	0.260	0.262	**0.324**	0.278	0.266	0.294	0.249	0.256	0.258
CS	0.675	0.697	0.720	0.713	0.706	**0.724**	0.643	0.672	0.674

Table 4.15: Results on documents with both topics using gold standard EDUs. LR: LexRank, LR-F: LexRank with Feng Hirst topics, LR-T: LexRank with treebank topics, TR: TextRank, TR-F: TextRank with Feng Hirst topics, TR-T: TextRank with treebank topics, SB: SumBasic, SB-F: SumBasic with Feng Hirst topics, SB-T: SumBasic with treebank topics.

Looking at the results, in most cases, using some form of RST topics improves performance. For LexRank, there is a clear progression in performance. The lowest scores are found when no topics are used. There is some improvement when the parser topics are used, and finally the best performance is achieved when the manually-annotated RST Treebank topics are used. These results illustrate that RST topics improve summarization performance. As the topics in the Treebank are likely more accurate representations of the texts' structure, using them results in the greatest increases in performance. On the other hand, the topics produced by the automatic parser are likely to be less accurate but still capture some of the relevant structure. Therefore, those topics improve performance compared to not using topics, but the increases are smaller than when using Treebank topics. The results are not as clear for TextRank. On all measures, using the parser topics decreases performance from the baseline of not using topics, while for all measures except ROUGE-2, the Treebank topics improve performance. For SumBasic, using either kind of topics increases the values of each of the measures. For both ROUGE measures, the parser topics actually outperform the Treebank topics, while the Treebank topics score better on unit overlap and cosine similarity. Some of the

differences are fairly small, but they suggest that even the parser topics can improve performance and in some cases they can perform as well as the Treebank topics. Although this is a small dataset, these results suggest a trend for parser topics to contribute slight improvements to performance while Treebank topics generally perform better.

I also performed the same comparison looking at topics found by the parser when it first performed its own EDU segmentation. The dataset included documents that have topics found by the parser in this way as well as topics in the Treebank and extractive summaries. This dataset is small like the previous one, with 17 documents. The following table shows the results from this second version of the RST parser in which it first performs EDU segmentation and then uses those EDUs to parse the text into relations.

	LR	LR-F	LR-T	TR	TR-F	TR-T	SB	SB-F	SB-T
R1	0.484	0.455	0.532	**0.574**	0.508	0.572	0.412	0.417	0.431
R2	0.290	0.272	0.356	**0.429**	0.349	0.401	0.196	0.219	0.230
UO	0.236	0.237	**0.268**	0.259	0.246	0.263	0.227	0.230	0.231
CS	0.647	0.627	0.667	**0.693**	0.673	0.691	0.611	0.629	0.633

Table 4.16: Results on documents with both topics and automatically parsed EDUs. LR: LexRank, LR-F: LexRank with Feng Hirst topics, LR-T: LexRank with treebank topics, TR: TextRank, TR-F: TextRank with Feng Hirst topics, TR-T: TextRank with treebank topics, SB: SumBasic, SB-F: SumBasic with Feng Hirst topics, SB-T: SumBasic with Treebank topics.

These results show that parser topics using parser EDUs do not increase performance as much as using gold standard EDUs. For LexRank, parser topics decrease the values for all measures except unit overlap, which has a similar value to not using topics. In contrast, the Treebank topics improve performance. For TextRank, parser topics decrease performance, and Treebank topics also decrease performance for most measures, although to a lesser extent. SumBasic shows a progression with parser topics increasing the scores from not using topics and Treebank topics increasing scores even more.

Overall, these results suggest that parser topics are less useful for summarization when they rely on automatic EDU segmentation and less useful than topics from the RST Treebank.

An interesting issue to consider is how well automatic parsers perform at finding topic relations in particular. Although there are many relation types in RST, this topic method specifically uses topic relations. Therefore, what is of the most importance for this method is whether the parsers accurately find and label topic relations. A few papers report results for parser performance by relation, which allows comparison of performance on topic relations to other relations. Hernault et al. (2010) report how well their parser retrieves the correct relation label. For topic change relations, which include topic shift and topic drift, they achieve a precision of 83%, recall of 39%, and f-score of 53%. For comparison, the highest values of these scores for any relation were 100%, 97%, and 95%. The best performance across these measures, given by the f-score of 95%, was found for attribution relations, which capture direct or indirect speech by connecting the speech to the source of the attribution. At the other end of the spectrum, the lowest values for precision, recall, and f-score were 31%, 2%, and 4%. Looking at the f-score, cause relations had the worst performance. These relations connect a situation to its cause or connect a cause and a result. In terms of f-score, topic relations have a score about halfway between the relations with the best and the worst scores. This suggests that classification of topic relations could be improved, particularly for these relations to be useful for other tasks. Looking at the results reported by Feng and Hirst (2014) for the second version of their parser indicates an even stronger need to improve classification of topic relations. They report that their model did not retrieve any instances of topic change

relations or textual organization relations as well as very low performance for a couple other relation types. They note the infrequency of those relations as well as their more abstract nature as reasons for the poor performance. These results demonstrate that automatic parsers struggle with identifying topic relations more than other relations. This provides more explanation for why the output of the parsers is not as useful when used for summarization. If the parsers do not reliably find and label topic relations, the topics they produce cannot be expected to be as useful as gold standard topic divisions. Further work is necessary to improve parsers, specifically their performance on less common relations such as topic change. As parsers improve, the output they produce can be used for more downstream tasks. Another area for additional research is how to discover RST topics in text without a full RST annotation. Since RST annotation is a difficult task, it would be useful to simplify the process by only finding topic relations. Further work is needed to determine whether such simplifications are possible and how they would be performed.

5.3 Summary of Findings

These experiments using an automatic RST parser demonstrate some interesting findings about how well these parsers perform and how the approach of using RST topics could be applied to any given document without manual RST annotations. One finding was that the topics found by the parser do not exactly align with the topics in the Treebank for the same documents. In addition, there were also differences in the EDU segmentations produced by the parser and the gold standard segmentations. The previous experiments in the chapter showed the effectiveness of using the topics in the Treebank as part of a summarization system. Therefore, RST topics are useful for this task. Even

though the parser does not perfectly find the same topics as in the Treebank, it is possible that the parser's topics, which are still based on RST, will also be useful for the task. The results confirmed this possibility. In several cases, performance improved when the parser's topics were used. This provides additional confirmation of the utility of RST topic structure for summarization and demonstrates that manual annotation of texts is not necessarily required in order to use RST information. On the other hand, several findings illustrate the shortcomings of using an automatic RST parser. Although improvements in performance are seen with the parser topics compared to not using topics, the increases are not as great or as consistent across summarizers as the improvements with the Treebank topics. At this point, the parser topics represent an intermediate step between not using topics and using gold standard topics. A similar issue is the difference between the two versions of the parser, one with gold standard EDU segmentation and the other with the parser's EDU segmentation. The parser with the gold standard segmentation performs better than the one that relies on its own segmentation. This shows that the output of the EDU segmenter is not the same as the gold standard, and these differences affect the topics that are produced. As more steps in the process are automated, the performance decreases so that using topics does not necessarily result in an improvement. The completely automated version represents how this method could be applied to any document, not in the Treebank. The inconsistent results with small or no increases in performance when using the automated version show that advances are needed to improve RST parsers and be able to apply the RST topic method to any document. The positive results found with the Treebank topics suggest that RST information is useful,

and as parsers improve and find more accurate RST structures, those structures can be used for tasks such as summarization.

6 Finding Topics in Other Documents

Not all documents in the RST Discourse Treebank include topic relations in their annotations. There are several possible reasons for the lack of topics in the other documents. Perhaps different annotators were more or less likely to use topic relations. It is also possible that there are no clear topic divisions in those documents or that the interaction of topics with the other RST relations is more complicated. This section explores the question of whether the documents without labeled topic relations do contain any clear topic divisions, and if so whether it is possible to use those divisions for summarization.

Sampling from the documents without topics shows that there is variation in whether they contain topics or not. Some documents are very short, making it difficult to identify multiple topics. While all of the documents are WSJ articles, there is some variation in the writing style used. For example, some documents are written more in the style of a letter. In general, it was more difficult to identify topics in documents of that style because they tended to include less overall structure. Another challenge in identifying topics comes from the scale of topics under consideration. The annotated topics tend to separate larger sections of text in which all of the sentences in one section discuss the same topic and all of the sentences in the other section discuss a different topic. However, in many of the documents without topic relations, it is hard to identify larger consistent topics beyond the topics at the sentence level. The sentence level topics gradually change throughout a document without the presence of clear topic divisions

between larger sections of text. Therefore, in many documents without topic relations, it is difficult to identify topic divisions.

In other documents, it is possible to find potential topic boundaries. In these cases, although the documents are not coded for topics in the RST annotation, looking at the documents indicates that there are changes in topic between sections. Since these documents are annotated with relations other than topic change relations, it is interesting to consider the types of relations used in those instances. These types include elaboration, contrast, and list. Elaboration involves providing additional information about a subject under discussion. In these cases, the presentation of additional information can appear like the introduction of a new topic, or perhaps a new subtopic. Contrast relations also have some similarity to topic relations. They connect two sections of text with different subject matter. However, contrast relations specifically involve the presentation of opposing information while topic relations can simply signal a change in the focus of the subject matter. Another relation that annotators used in cases that could potentially be considered topic changes is the list relation. As its name would suggest, the list relation involves a set of elements that share some type of parallel structure or parallel relationship. Particularly when the list elements are larger sections of text beyond a single sentence, these sections resemble distinct topics and could potentially be separated by topic boundaries even though they are not annotated with topic relations. These relation types are all examples of the types of relations that were present in the annotations in instances where there seem to be different topics. As discussed, these relations all have qualities that make them similar to topic changes.

The following example illustrates the similarity between list relations and topic relations. The text is divided with brackets according to the potential topic boundaries placed in the document.

> [Elcotel Inc. expects fiscal second-quarter earnings to trail 1988 results, but anticipates that several new products will lead to a "much stronger" performance in its second half.
>
> Elcotel, a telecommunications company, had net income of $272,000, or five cents a share, in its year-earlier second quarter, ended Sept. 30.
> Revenue totaled $5 million.
> George Pierce, chairman and chief executive officer, said in an interview that earnings in the most recent quarter will be about two cents a share on revenue of just under $4 million.
>
> The lower results, Mr. Pierce said, reflect a 12-month decline in industry sales of privately owned pay telephones, Elcotel's primary business.
> Although Mr. Pierce expects that line of business to strengthen in the next year, he said Elcotel will also benefit from moving into other areas.]
>
> [Foremost among those is the company's entrance into the public facsimile business, Mr. Pierce said.
> Within the next year, Elcotel expects to place 10,000 fax machines, made by Minolta in Japan, in hotels, municipal buildings, drugstores and other public settings around the country.
> Elcotel will provide a credit-card reader for the machines to collect, store and forward billing data.
>
> Mr. Pierce said Elcotel should realize a minimum of $10 of recurring net earnings for each machine each month.]
>
> [Elcotel has also developed an automatic call processor that will make further use of the company's system for automating and handling credit-card calls and collect calls.
> Automatic call processors will provide that system for virtually any telephone, Mr. Pierce said, not just phones produced by Elcotel.
>
> The company will also be producing a new line of convenience telephones, which don't accept coins, for use in hotel lobbies, office lobbies, hospitality lounges and similar settings.
>
> Mr. Pierce estimated that the processors and convenience phones would produce about $5 of recurring net earnings for each machine each month.]

In this text, the first topic introduces the main idea, and the second and third topics each provide examples. There is clearly a sense in which these sections represent different topics even though there are no topic relations in this text's RST Treebank annotation. In that annotation, the second and third topics are related to each other through a list relation, and the combination of these two topics is related to the first topic through an example relation. Therefore, there is similarity in these relation types, and it seems possible to consider some of these other relations as also representing topic changes.

In order to test whether manually-derived topics that are not annotated with topic relations in the Treebank are useful for summarization, I found a set of documents that contain potential topics and divided them into topics accordingly. These documents were sampled from the set of documents that have corresponding gold standard extractive summaries in the Treebank. The resulting set contains 22 documents. This set is relatively small, and is therefore intended as an exploration of whether similar topics can be found without annotated topic relations and the effects of these topics on summarization. The same process for summarization and evaluation that was used in the other experiments was used to summarize these documents. The results are presented below.

	LR	LR-T	TR	TR-T	SB	SB-T
Avg ROUGE-1	0.493	0.517	**0.550**	0.504	0.441	0.372
Avg ROUGE-2	0.315	0.345	**0.395**	0.323	0.251	0.179
Avg Unit Overlap	0.263	**0.299**	0.275	0.267	0.254	0.219
Avg Cos Similarity	0.664	**0.691**	0.679	0.675	0.624	0.602

Table 4.17: Results with manually-segmented topics in other documents

The table includes the results without any topics and the results using the manually-derived topics. The results differ depending on which summarizer is used. These topics improve performance with LexRank but decrease performance with TextRank and

SumBasic. Between TextRank and SumBasic, the topics decrease performance the most with SumBasic. Looking at the highest values for each measure shows an interesting result. For the ROUGE measures, the best values are achieved by a model without topics, specifically TextRank. On the other hand, for unit overlap and cosine similarity, the best values are achieved by a model with topics, specifically LexRank. Therefore, if choosing the best overall model, the results are divided over whether a model with or without topics should be chosen.

Overall, these results suggest that these types of topics improve summarization performance in some cases, but the results are not as consistent as they are with annotated topic relations. There is a strong interaction with which summarizer is used, with the effect of topics dependent on the summarizer. Understanding the specific qualities of the summarizers and how they interact with the divisions created by these topics requires additional research and is an area for future work. Another area for exploration is the similarities and differences between topic relations and the relations used in the RST annotations of these documents. It is possible that topic relations are capturing a different type of information than other relations, such as contrast and list, that makes them more useful for achieving better information coverage in a summary. Understanding whether other relations could be used to approximate topic divisions would be useful for expanding how RST can be used for topic segmentation and topic-based summarization.

7 Connecting Summarization to Compression

Chapter 2 described the similarity between the tasks of summarization and compression. Both seek to condense information into a smaller form. Text compression typically operates at the string level by finding repeated characters. On the other hand,

summarization works at the meaning level by finding pieces of text that convey similar meanings and reducing redundancy in summaries. Given that the results in this chapter show the utility of topics based on RST for summarization, this section explores the connection between these topics and compression to see whether any features of these topics correlate with compressibility.

Chapter 2 discussed how compression could be used to produce a measure of how similar or dissimilar two pieces of text are. Simovici et al. (2015) propose a method for calculating dissimilarity of two documents based on the ratio between the size when two documents are concatenated and then compressed and the sum of the sizes when the two documents are separately compressed. This is shown in the following equation, in which $C(x)$ represents the size of the compressed text.

(4.8) $dissimilarity(x, y) = \frac{C(xy)}{C(x)+C(y)}$

For identical documents, the dissimilarity value would be around 0.5 because the compression algorithm makes use of the fact that the two halves of the concatenated document are the same. Therefore, documents that are more similar have values closer to 0.5, and more dissimilar documents have values closer to 1. This measure captures the idea that if two documents are similar, they will share common elements and a compression algorithm will be able to reduce them more than two dissimilar documents which share less for the algorithm to reduce.

Comparing pieces of text in terms of their dissimilarity relates to the idea of dividing texts into topics. Sentences within a topic should be similar to each other, but different topics should be dissimilar. Using topics is a way to ensure coverage of all distinct concepts in a text. For topics to be effective for this purpose, they should ideally

be as dissimilar to each other as possible. If two topics are very similar, then including coverage of both topics in a summary is likely to produce some redundancy or prevent a more distinct topic from being included in the summary's limited space. If topics are dissimilar to each other, they can better capture the full range of ideas in the text and summarizing those topics will result in more complete summaries.

7.1 Experiments using Dissimilarity based on Compression

Based on these ideas, I explored the dissimilarity of the RST topics within a document. The same set of 51 documents that was used in the previous experiments was used. Compression was performed using the gzip algorithm described in Chapter 2. Each document was considered one at a time. Looking at the topics in a document, all pairwise dissimilarity comparisons were done. Each topic was compared to every other topic in the document. The topics were compressed individually, and their concatenation was compressed in order to calculate dissimilarity. When all of the pairwise comparisons were completed, the dissimilarity scores were averaged to find an average dissimilarity for the document. To calculate an overall score for the dataset, the document scores were averaged to produce a dissimilarity value for the dataset.

Similar to the previous experiments, random topics were used as a baseline for comparison. Using the same process to determine random topics of the same size as the RST topics, the texts were randomly divided into topics 10 times. With these topics, the dissimilarity calculations were performed as above. A value for dissimilarity of the dataset according to the random topics was produced. To compare with the RST value, the mean and standard deviation of the 10 random runs were calculated. The results are presented in the following table.

Mean	0.9152
Standard Deviation	0.0003
RST Topics	0.9244

Table 4.18: Mean and standard deviation of dissimilarity of random topics compared to RST topics

The dissimilarity value when using RST topics is significantly higher than when using random topics. The difference is small, at around only 1%, but looking at the standard deviation, the difference is significant as it is several standard deviations above the mean. Higher values mean that the topics are more dissimilar. The interpretation of these results is that topics determined by RST relations are more dissimilar than randomly determined topics.

Returning to the suggestions above, topics that are more dissimilar should better capture the range of information present in the text. Since it is important for summaries to include this range, dividing a text into dissimilar topics should be more useful for summarization than using similar topics. In the previous experiments, RST topics were shown to improve summarization performance more than random topics. This exploration of dissimilarity showed that the texts of RST topics tend to be more dissimilar to each other than those of random topics. Therefore, there is some evidence that dissimilar topics are more useful for summarization than similar topics. This agrees with the intuition that including coverage of all distinct topics in a text is crucial for a good summary. Dividing a text into dissimilar topics separates information into different groups with less redundancy between topics. Performing summarization over these topics then allows each distinct topic to be included in a summary while preventing information from being repeated or any one topic receiving too much coverage. This experiment with

dissimilarity highlights an interesting quality of RST topics and provides additional support for the use of RST topics in summarization.

This exploration also illustrates one way that compression can be used in relation to summarization. There is a clear parallel between the two tasks, but it is less obvious how the information given by compression can be directly used to influence summarization beyond providing a useful analogy. These results suggest that compression and the compressed sizes of texts in particular could be a useful part of deciding how texts should be divided into topics. In this case, the dissimilarity found through compression provides some support and a possible explanation for why RST topics are useful for summarization. On the other hand, there is potential for compression to be used as part of the topic division process by helping decide between proposed topic divisions or even by using the dissimilarity information to suggest possible divisions. Further research is necessary to determine the effectiveness of using compression information in this way, but these preliminary results suggest that it could be a useful component of the topic division and summarization process.

8 Conclusion

This chapter presented experiments exploring the use of topics for summarization. RST topics were shown to improve summarization performance compared to not using topics or using random topics. RST topics were also compared to a notion of topic based on word distributions using Latent Semantic Analysis. RST topics outperformed LSA topics in all cases, with LSA topics generally not improving performance compared to not using topics. The final part of this chapter explored how the use of RST topics could possibly be extended to more data by using an automatic RST parser to annotate texts.

The results showed that topics found by RST parsers can be useful for summarization, although not as useful as gold standard topics. The small improvements suggest that RST topics are indeed useful for summarization and have the potential to be more widely used as parsers continue to improve.

Chapter 5

Conclusion

1 Research Questions

Several questions introduced in the first chapter guided the work in this dissertation. This chapter returns to those questions and describes how the research conducted addressed those questions. This conclusion summarizes the research and results. The contributions of this work and suggestions for future directions of this work are also discussed.

One of the questions mentioned in the introduction is how to determine the ideal length for a summary. This relates to text compression, which was discussed in chapter two. Just as compression takes a string and reduces it to the minimum necessary information to reconstruct the text, summarization should reduce a document to its core of crucial information to be able to understand the original document from the summary. If this information could be determined automatically and redundancy at the meaning level could be determined in a principled way like redundancy at the character level, then the issue of summary length would also be solved. A summary would only be as long as it needed to be to convey the necessary information without redundancy. The desire to understand the ideal length for a summary of a given text as well as the overall goal of creating summaries with good information coverage motivates thinking of summarization as similar to compression. Chapter two explored this connection between summarization

and compression to see whether compression algorithms can be useful for summarization. The compressibility of a text was compared to text features including the percentage of unique words contained in the text. This analysis showed that the amount a text can be compressed is very correlated with the presence of unique words, reducing the patterns found by compression algorithms to the word level, with features such as word order having little effect. These explorations confirmed an interesting correlation between text compression and information at the word level, and they motivate taking the ideas of compression beyond the word level to use these ideas for summarization.

Chapter four returned to the connection between compression and summarization. It explored the use of compression information as a measure of dissimilarity between sections of text. Specifically, dissimilarity was calculated between the topics of a document based on the idea that dividing a document into topics that are more dissimilar should be more useful for summarization. Dissimilar topics are more likely to capture the full range of information present in the document, and therefore using those topics should be more useful for summarization. The results of the dissimilarity comparison showed that there was a small but significant difference between the dissimilarity of RST topics and random topics. This finding suggests that one reason RST topics improve summarization is because they help to ensure that a summary contains coverage of all of the distinct concepts in a text. This relates to the other questions discussed in the introduction.

The questions introduced in the first chapter also include how to represent the information in a document so that the best information can be selected for a summary. How to determine what is the best information is related to another question of how to

achieve broad coverage of the information in the text, with the idea that a summary should cover all major concepts in the text with emphasis on including some coverage of all concepts rather than detailed coverage of any particular concept. These questions led to the discussion of topics and the use of topic structure for summarization. In order to ensure coverage of all of the important concepts in a summary, texts can be divided into topics and sentences selected to represent each topic in the summary. One important question is how topics are defined. Chapter three described different notions of what it means to be a topic. The discussion focused on two notions of topic in particular: one defined using Rhetorical Structure Theory and the other defined using distributional topic modeling methods such as Latent Semantic Analysis. Chapter three also included discussion of the differences between these two notions of topic. Latent Semantic Analysis uses mathematical methods to transform the words in a document into a representation of meaning based on their distribution. The method finds topics and defines words according to their relations to each of the topics. On the other hand, topics in Rhetorical Structure Theory are defined by relations that connect sections of text when a topic change has occurred. In thinking about these different notions of topic, one important issue to consider is the distinction between a definition of topic and the methods that can be used to automatically determine those topics. Chapter three discussed both of these issues to understand the type of information captured by different topics as well as how to implement these notions of topic into an automatic system.

2 Summary of Results

To address the research questions and explore them in a formal way, experiments were conducted to explore how topics can be used to improve automatic summarization.

These experiments are described in chapter four. Experiments were conducted in a modular nature to allow for direct comparison of the results when using topics or not using topics. Specifically, several previous summarizers were tested to see how the results were affected by the use of topics. The overall process involved dividing a text into topics, summarizing each topic, and combining the summaries of each topic into a final summary of the entire text. In this way, topics were treated as independent components of a text that each contributes to the understanding of the document and should be represented in the summary. The results of incorporating topics were compared to the results when summarizing the entire document without dividing it into topics. This process allows for direct comparison of using topics and not using topics as well as comparison between different notions of topic as the topic divisions can be changed and used with the same summarizers. Using the same basic process, several different factors were explored. Three different summarizers were used to confirm that the results are not limited to any specific summarizer. The two notions of topic described above, based on RST and LSA, were both tested, as well as topics formed from randomly dividing sentences into groups. The random topics serve as a baseline to see whether an improvement from the use of topics could simply be explained by dividing a text into smaller sections regardless of whether those sections are linguistically motivated. Different values of the summarization percentage were also tested. The summarization percentage refers to how much of the original text is retained for a summary. Values from 10% to 40% were tested. Lower percentages result in shorter summaries, meaning that a summarizer must be precise in order to select the same information that is contained in the reference summary. Higher percentages result in longer summaries with more

opportunities to overlap with the reference summary but also more chances to include redundant or superfluous information. All of these different factors were tested to gain a broad understanding of how topics interact with summarization. In addition, several different measures were used to evaluate the performance of the summarization system under these different conditions.

The experiments produced several interesting results. The results of using RST topics compared to no topics showed the positive effect of topic structure. In almost all cases, for different values of each of these variables, the highest performance resulted from summarizing using topics. Therefore, topics clearly have a positive impact on performance regardless of the specifics of which summarizers or summarization percentages are used. However, there are also some interesting effects when looking at different values of these variables. For example, the increases in performance are greater when the summarization percentage is lower, meaning the summaries are smaller. This result suggests that information from topic structure is most useful when summary space is the most limited and the choices of what to include are the most important.

The improvements from RST topics were confirmed when the results were compared to the results of using random topics. The comparison with random topics was intended to check whether summarization performance could be improved simply by summarizing smaller sections of text or whether the improvements are due to meaningful qualities of the RST topics. The results showed that RST topics performed significantly better than random topics. This confirms that the improvement seen with RST topics is not simply due to dividing the text into smaller sections before summarization.

Another experiment looked at the effect of a different notion of topic, one based on LSA. The results showed that topics formed from LSA information did not improve summarization performance as much as RST topics. The difference in performance was consistent across summarizers, evaluation measures, and dimensions in the LSA model. In addition, compared to not using any topics, LSA topics did not consistently increase performance, with small increases in some cases but decreases in other cases. The general finding was that the types of topics found by topic modeling methods such as LSA are not very useful for summarization. As discussed, while these methods can successfully distinguish between documents with very different topics, it is likely that topics within a single document are more similar to each other than the topics of distinct documents. The lack of improvement for single-document summarization seen with LSA topics is likely due to these differences in the type and granularity of topics within documents compared to topics across a collection of documents.

To be able to extend the improvements found with RST topics to a wider set of documents, chapter four also described experiments with an automatic RST parser. Using an automatic parser would remove the need for manual annotation of RST relations. For a direct comparison of whether the topic annotations produced by a parser are as useful as manual annotations from the RST Treebank, the same documents from the Treebank were given as input to an automatic parser. The results showed that topics created by the parser produced some improvements compared to not using topics. However, the improvements were not as great or as consistent across summarizers as manually annotated topics. One problem with using automatic parsers is that they can introduce errors that affect downstream tasks, including summarization in this case. Improving

automatic RST parsers is an active area of research, but current parsers do not match human performance, and as topic relations are higher-level, more abstract relations, automatic parsers tend to perform worse on those relations than others. More work is necessary to improve these parsers so that they can reliably be used instead of manual annotation. As parsers improve, their output should become more useful for tasks like summarization.

3 Contributions of This Dissertation

This dissertation made several contributions to the research area of automatic summarization. First, it showed that topics are a useful structure for summarization. Taking advantage of a text's organization into groups of related sentences was found to improve the selection of sentences for a summary. This result confirms the idea that summarization should focus on achieving coverage of all of the concepts in a text and shows that topics are an effective method to achieve this coverage. A related contribution comes from demonstrating the effectiveness of a straightforward modular method for incorporating topics into summarization. Topic structure was included in the summarization process by summarizing individual topics and combining the topic summaries to create a complete summary. The simplicity of this method makes it possible to test other summarizers and other notions of topic and directly compare the results. The fact that performance improvements were found with this method demonstrates that this is a straightforward but effective method to test the influence of topics.

This dissertation also explored a new way of using Rhetorical Structure Theory for summarization. Specifically, it used RST to define topics and divide documents into

topics. Exploring this notion of topic based on RST showed that this type of information is useful for summarization. This work focused on single-document summarization. The issue of interest was whether different topics within one document could be used to improve the coverage of a summary. Therefore, a notion of topic that operates within a document was necessary. RST finds relations within a document, and the topic relations identify the types of topic changes that should be useful for finding the distinct concepts in a text. The definition of topic based on RST was shown to improve automatic summarization performance, demonstrating both how topics can be useful for the task as well as another way of using RST information for summarization.

After demonstrating the effectiveness of RST topics, it was important to consider another notion of topic. Specifically, RST topics were compared to the types of topics found by topic modeling methods. Using the same method for incorporating topics into summarization, these other topics were tested. The results demonstrated that not all notions of topic are equally effective. This work showed that topic modeling methods are not very useful for distinguishing between similar topics. In particular, as mentioned above, single-document summarization requires finding topics within a document, which are likely to be more similar to each other than topics from different documents. The lack of improvement seen with the topic modeling methods is an important result because it shows that not all notions of topic are equally well-suited for summarization. Although topic modeling is a common way to conceptualize and work with topics, this work showed that there are limits on the kinds of tasks that these topics can be used for. The contrast between these topics and RST topics highlights the effectiveness of the RST topics.

4 Future Work

One area for additional research is how to discover RST topics in text without requiring a full RST annotation. Since RST annotation is a difficult task, it would be useful to simplify the process by only finding topic relations. The current work relied on topic relations to divide texts into topics. However, the other types of relations are not specifically used. As the texts used in these experiments were already annotated with all relations, no extra effort was needed to find and label relations that were not used. Further work is needed to determine whether such simplifications are possible and how they would be performed.

Another area for future work is expanding on the similarity between summarization and compression. As discussed in previous chapters, compression typically operates at the character level. Compression algorithms find redundancy by looking for repeated characters. While character repetition is very useful for basic text compression, the same idea could be extended to any other method for finding redundancy. What would be useful for summarization is to find and reduce redundancy at the meaning level. However, while character repetition is straightforward to determine, there is no simple way to determine whether two sections of text have the same meaning. That would require translating a text into a representation of its meaning and comparing those representations. The challenge is to find a representation that adequately captures meaning and could be used to decide whether two sections of text convey the same information. In addition, there are challenges of learning the representation and comparing texts automatically, without requiring human judgment. This is a very interesting area for future research. Extending the method of compression beyond the

character level could change the way summarization is performed. It would also allow for interesting research into measuring the amount of distinct content in a text and help answer the question of how to determine the appropriate length for a summary automatically.

Bibliography

Abuobieda, Albaraa, Naomie Salim, Ameer Tawfik Albaham, Ahmed Hamza Osman, and Yogan Jaya Kumar. 2012. Text summarization features selection method using pseudo genetic-based model. In *Information Retrieval & Knowledge Management (CAMP), 2012 International Conference on* (pp. 193-197). IEEE.

Asher, Nicholas. 2004. Discourse topic. *Theoretical Linguistics*, *30*(2-3), pp.163-201.

Banko, Michele, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open Information Extraction from the Web. In *IJCAI* (Vol. 7, pp. 2670-2676).

Beeferman, Doug, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning*, *34*(1), pp.177-210.

Bellegarda, Jerome R. 2000. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, *88*(8), pp.1279-1296.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, *3*(Feb), pp.1137-1155.

Bird, Steven, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Blei, David M. 2012. Probabilistic topic models. *Communications of the ACM*, *55*(4), pp.77-84.

Bradford, Roger B. 2008. An empirical study of required dimensionality for large-scale latent semantic indexing applications. In *Proceedings of the 17th ACM conference on Information and knowledge management* (pp. 153-162). ACM.

Burton, N.G. and J.C.R. Licklider. 1955. Long-range constraints in the statistical structure of printed English. *The American journal of psychology*, *68*(4), pp.650-653.

Cardoso, Paula C.F., Maria L.R.C. Jorge, and Thiago A.S. Pardo. 2015. Exploring the Rhetorical Structure Theory for multi-document summarization. In *Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural, XXXI*. Sociedad Española para el Procesamiento del Lenguaje Natural-SEPLN.

Carlson, Lynn and Daniel Marcu. 2001. Discourse tagging reference manual. *ISI Technical Report ISI-TR-545*, *54*, p.56.

Carlson, Lynn, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current and new directions in discourse and dialogue* (pp. 85-112). Springer, Dordrecht.

Carlson, Lynn, Mary Ellen Okurowski, and Daniel Marcu. 2002. *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania.

Chafe, Wallace. 1994. Discourse, consciousness, and time. *Discourse*, *2*(1).

Chen, Bei, Jun Zhu, Nan Yang, Tian Tian, Ming Zhou, and Bo Zhang. 2016. Jointly Modeling Topics and Intents with Global Order Structure. In *AAAI* (pp. 2711-2717).

Cheng, Jianpeng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

Chengcheng, Li. 2010. Automatic text summarization based on rhetorical structure theory. In *International Conference on Computer Application and System Modeling (ICCASM)* (Vol. 13, pp. V13-595). IEEE.

Choi, Freddy Y. 2000. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference* (pp. 26-33). Association for Computational Linguistics.

Chopra, Sumit, Michael Auli, Alexander M. Rush, and S.E.A.S. Harvard. 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *HLT-NAACL* (pp. 93-98).

Christensen, Janara, Stephen Soderland, and Oren Etzioni. 2013. Towards Coherent Multi-Document Summarization. In *HLT-NAACL*, 1163-1173.

Conroy, John M., Judith D. Schlesinger, and Dianne P. O'Leary. 2006. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL on Main conference poster sessions*, 152-159. Association for Computational Linguistics.

Dagan, Ido, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment* (pp. 177-190). Springer, Berlin, Heidelberg.

Deerwester, Scott, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, *41*(6), p.391.

Document Understanding Conference. http://duc.nist.gov.

Du, Lan, John K. Pate, and Mark Johnson. 2015. Topic Segmentation with an Ordering-Based Topic Model. In *AAAI*(pp. 2232-2238).

Dumais, Susan T. 1994. Latent semantic indexing (LSI) and TREC-2. *Nist Special Publication Sp*, pp.105-105.

Dumais, Susan T. 1995. Latent semantic indexing (LSI): TREC-3 report. *Nist Special Publication SP*, pp.219-219.

Eisenstein, Jacob. 2009. Hierarchical text segmentation from multi-scale lexical cohesion. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 353-361). Association for Computational Linguistics.

Erkan, Günes and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, pp.457-479.

Feng, Vanessa Wei and Graeme Hirst. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1* (pp. 60-68). Association for Computational Linguistics.

Feng, Vanessa Wei and Graeme Hirst. 2014. A Linear-Time Bottom-Up Discourse Parser with Constraints and Post-Editing. In *ACL (1)* (pp. 511-521).

Ferreira, Rafael, Luciano de Souza Cabral, Rafael Dueire Lins, Gabriel Pereira Silva, Fred Freitas, George D. Cavalcanti, Ronaldo Lima, Steven J. Simske, and Luciano Favaro. 2013. Assessing sentence scoring techniques for extractive text summarization. *Expert systems with applications*, *40*(14), 5755-5764.

Foltz, Peter W. 1996. Latent semantic analysis for text-based research. *Behavior Research Methods*, *28*(2), pp.197-202.

Fournier, Chris. 2013. Evaluating text segmentation using boundary edit distance. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Vol. 1, pp. 1702-1712).

Genest, Pierre-Etienne and Guy Lapalme. 2012. Fully abstractive approach to guided summarization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2* (pp. 354-358). Association for Computational Linguistics.

Goldberg, Yoav. 2016. A Primer on Neural Network Models for Natural Language Processing. *J. Artif. Intell. Res.(JAIR)*, *57*, pp.345-420.

Goyal, Naman and Jacob Eisenstein. 2016. A Joint Model of Rhetorical Discourse Structure and Summarization. *EMNLP 2016*, p.25.

Grewal, Amardeep, Timothy Allison, Stanko Dimitrov, and Dragomir Radev. 2003. Multi-document summarization using off the shelf compression software. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5* (pp. 17-24). Association for Computational Linguistics.

Griffiths, Thomas L., Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. 2005. Integrating topics and syntax. In *Advances in neural information processing systems* (pp. 537-544).

Griffiths, Thomas L., Mark Steyvers, and Joshua B. Tenenbaum. 2007. Topics in semantic representation. *Psychological review*, *114*(2), p.211.

Griggs, Julian S. 2015. TL; DR: Automatic Summarization With Textual Annotations.

Gundel, Jeanette K. 1988. Universals of topic-comment structure. *Studies in syntactic typology*, *17*, pp.209- 239.

Gundel, Jeanette K. and Thorstein Fretheim. 2004. Topic and focus. *The handbook of pragmatics*, *175*, p.196.

Guo, Weiwei and Mona Diab. 2012. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-volume 1* (pp. 864-872). Association for Computational Linguistics.

Gupta, Pankaj, Vijay Shankar Pendluri, and Ishant Vats. 2011. Summarizing text by ranking text units according to shallow linguistic features. In *Advanced Communication Technology (ICACT), 2011 13th International Conference on* (pp. 1620-1625). IEEE.

Hearst, Marti A. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics*, *23*(1), pp.33-64.

Hernault, Hugo, Helmut Prendinger, and Mitsuru Ishizuka. 2010. HILDA: A discourse parser using support vector machine classification. *Dialogue & Discourse*, *1*(3).

Hyona, Jukka, Robert F. Lorch, and Johanna K. Kaakinen. 2002. Individual differences in reading to summarize expository text: Evidence from eye fixation patterns. *Journal of Educational Psychology*, *94*(1), pp.44-55.

Ji, Yangfeng and Jacob Eisenstein. 2014. Representation Learning for Text-level Discourse Parsing. In *ACL (1)* (pp. 13-24).

Johnston, Peter and Peter Afflerbach. 1985. The process of constructing main ideas from text. *Cognition and Instruction*, *2*(3-4), pp.207-232.

Kågebäck, Mikael, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. 2014. Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)@ EACL* (pp. 31-39).

Kaynar, Oğuz, Yasin Görmez, Yunus Emre Işık, and Ferhan Demirkoparan. 2017. Comparison of graph based document summarization method. In *Computer Science and Engineering (UBMK), 2017 International Conference on* (pp. 598-603). IEEE.

Kibby, Michael W. 1980. Intersentential processes in reading comprehension. *Journal of Reading Behavior*, *12*(4), pp.299-312.

Lambrecht, Knud. 1996. *Information structure and sentence form: Topic, focus, and the mental representations of discourse referents* (Vol. 71). Cambridge University Press.

Landauer, Thomas K. and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*(2), p.211.

Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, *25*(2-3), pp.259-284.

Lee, Heeyoung, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In *Proceedings of the fifteenth conference on computational natural language learning: Shared task* (pp. 28-34). Association for Computational Linguistics.

Li, Hang. 2017. Deep learning for natural language processing: advantages and challenges. *National Science Review*.

Li, Junyi Jessy, Kapil Thadani, and Amanda Stent. 2016. The Role of Discourse Units in Near-Extractive Summarization. In *SIGDIAL Conference* (pp. 137-147).

Lin, Chin-Yew. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop* (Vol. 8).

Lin, Chin-Yew and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 71-78). Association for Computational Linguistics.

Liu, Fei, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations.

Lloret, Elena and Manuel Palomar. 2009. A gradual combination of features for building automatic summarisation systems. In *Text, Speech and Dialogue* (pp. 16-23). Springer Berlin Heidelberg.

Lorch, Robert F., Elizabeth Pugzles Lorch, and Patricia D. Matthews. 1985. On-line processing of the topic structure of a text. *Journal of memory and language*, *24*(3), pp.350-362.

Lorch Jr, Robert F., Elizabeth Pugzles Lorch, and Ann M. Mogan. 1987. Task effects and individual differences in on-line processing of the topic structure of a text. *Discourse Processes*, *10*(1), pp.63-80.

Louis, Annie, Aravind Joshi, and Ani Nenkova. 2010. Discourse indicators for content selection in summarization. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue* (pp. 147-156). Association for Computational Linguistics.

Luhn, Hans Peter. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, *2*(2), 159-165.

Mann, William C. and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, *8*(3), pp.243-281.

Marcu, Daniel. 1999. Instructions for manually annotating the discourse structures of texts. *Unpublished manuscript, USC/ISI*.

Marcu, Daniel. 2000. *The theory and practice of discourse parsing and summarization*. MIT press.

Marcus, Mitchell, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. 1999. Treebank-3 LDC99T42. Web Download. Philadelphia: Linguistic Data Consortium.

Mei, Qiaozhu, Jian Guo, and Dragomir Radev. 2010. Divrank: the interplay of prestige and diversity in information networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1009-1018). Acm.

Mihalcea, Rada and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *EMNLP* (Vol. 4, pp. 404-411).

Mittal, Namita, Basant Agarwal, Nikita Vijay, and Adarsh Gupta. Text Summarization with Semantics Information.

Moawad, Ibrahim F. and Mostafa Aref. 2012. Semantic graph reduction approach for abstractive Text Summarization. In *2012 Seventh International Conference on Computer Engineering & Systems (ICCES),* 132-138. IEEE.

Moore, Johanna D. and Martha E. Pollack. 1992. A problem for RST: The need for multi-level discourse analysis. *Computational linguistics*, *18*(4), pp.537-544.

Morris, Andrew H., George M. Kasper, and Dennis A. Adams. 1992. The effects and limitations of automated text condensing on reading comprehension performance. *Information Systems Research*, *3*(1), pp.17-35.

Moser, Megan and Johanna D. Moore. 1996. Toward a synthesis of two accounts of discourse structure. *Computational linguistics*, *22*(3), pp.409-419.

Murdock, Vanessa Graham. 2006. *Aspects of sentence retrieval* (Doctoral dissertation, University of Massachusetts Amherst).

Nallapati, Ramesh, Feifei Zhai, and Bowen Zhou. 2017. SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. *hiP (yi= 1| hi, si, d)*, *1*, p.1.

Nallapati, Ramesh, Bowen Zhou, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv pre- print arXiv:1602.06023*.

Nenkova, Ani, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing (TSLP)*, *4*(2), p.4.

Nenkova, Ani and Lucy Vanderwende. 2005. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, *101*.

Pang, Bo and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics* (p. 271). Association for Computational Linguistics.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and Jake Vanderplas. 2011. Scikit- learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), pp.2825-2830.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Pevzner, Lev and Marti A. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, *28*(1), pp.19-36.

Radev, Dragomir R., Timothy Allison, Sasha Blair-Goldensohn, John Blitzer, Arda Celebi, Stanko Dimitrov, Elliott Drabek, Ali Hakim, Wai Lam, Danyu Liu, Jahna Otterbacher, Hong Qi, Horacio Saggion, Simone Teufel, Michael Topper, Adam Winkel, and Zhu Zhang. 2004. MEAD-A Platform for Multidocument Multilingual Text Summarization. In *LREC*.

Rehurek, Radim and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*.

Riedl, Martin and Chris Biemann. 2012. TopicTiling: a text segmentation algorithm based on LDA. In *Proceedings of ACL 2012 Student Research Workshop* (pp. 37-42). Association for Computational Linguistics.

Robertson, Stephen. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of documentation*, *60*(5), 503-520.

Rush, Alexander M., Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Séaghdha, Diarmuid O. and Simone Teufel. 2014. Unsupervised learning of rhetorical structure with un-topic models. In *COLING* (pp. 2-13).

See, Abigail, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. *arXiv preprint arXiv:1704.04368*.

Shannon, Claude E. 1951. Prediction and entropy of printed English. *Bell Labs Technical Journal*, *30*(1), pp.50-64.

Simovici, Dan A., Tong Wang, Ping Chen, and Dan Pletea. 2015. Compression and data mining. In *Computing, Networking and Communications (ICNC), 2015 International Conference on* (pp. 551-555). IEEE.

Song, Wei and Soon Cheol Park. 2007. A novel document clustering model based on latent semantic analysis. In *Semantics, Knowledge and Grid, Third International Conference on* (pp. 539-542). IEEE.

Sparck Jones, Karen. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, *28*(1), 11-21.

Steinberger, Josef and Karel Ježek. 2012. Evaluation measures for text summarization. *Computing and Informatics*, *28*(2), pp.251-275.

Sumy Python library. https://github.com/miso-belica/sumy.

Thompson, Sandra A. and William C. Mann. 1987. Rhetorical structure theory. *IPRA Papers in Pragmatics*, *1*(1), pp.79-105.

Tokunaga, Takenobu and Iwayama Makoto. 1994. Text categorization based on weighted inverse document frequency. In *Special Interest Groups and Information Process Society of Japan (SIG-IPSJ)*.

Vanderwende, Lucy, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, *43*(6), pp.1606-1618.

Van Dijk, Teun A. 1977. Sentence topic and discourse topic. *Papers in Slavic Philology*, *1*, pp.49-61.

Van Kuppevelt, Jan. 1995. Discourse structure, topicality and questioning. *Journal of linguistics*, *31*(1), pp.109- 147.

Varadarajan, Ramakrishna and Vagelis Hristidis. 2006. A system for query-specific document summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management* (pp. 622-631). ACM.

Wei, Chih-Ping, Christopher C. Yang, and Chia-Min Lin. 2008. A Latent Semantic Indexing-based approach to multilingual document clustering. *Decision Support Systems*, *45*(3), pp.606-620.

Welch, Terry A. 1984. A technique for high-performance data compression. *Computer*, *6*(17), pp.8-19.

Witten, Ian H., Timothy C. Bell, Alistair Moffat, Craig G. Nevill-Manning, Tony C. Smith, and Harold Thimbleby. 1994. Semantic and generative models for lossy text compression. *The Computer Journal*, *37*(2), pp.83-87.

Wong, Kam-Fai, Mingli Wu, and Wenjie Li. 2008. Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1* (pp. 985-992). Association for Computational Linguistics.

Xie, Shasha and Yang Liu. 2008. Using corpus and knowledge-based similarity measure in maximum marginal relevance for meeting summarization. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on* (pp. 4985-4988). IEEE.

Yaari, Yaakov. 1997. Segmentation of expository texts by hierarchical agglomerative clustering. *arXiv preprint cmp-lg/9709015*.

Yang, Yinfei, Forrest Sheng Bao, and Ani Nenkova. 2017. Detecting (Un) Important Content for Single-Document News Summarization. *arXiv preprint arXiv:1702.07998*.

Yih, Wen-tau, Joshua Goodman, Lucy Vanderwende, and Hisami Suzuki. 2007. Multi-Document Summarization by Maximizing Informative Content-Words. In *IJCAI*, 1776-1782.

Zelikovitz, Sarah and Haym Hirsh. 2001. Using LSI for text classification in the presence of background text. In *Proceedings of the tenth international conference on Information and knowledge management* (pp. 113-118). ACM.

Zhang, Hui, Xueliang Zhang, and Guanglai Gao. 2015. Document summarization based on semantic representations. In *Asian Language Processing (IALP), 2015 International Conference on* (pp. 152-155). IEEE.

Ziv, Jacob and Abraham Lempel. 1977. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, *23*(3), pp.337-343.

Ziv, Jacob and Abraham Lempel. 1978. Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, *24*(5), pp.530-536.