uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Darren Kipp

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S.

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Shallow Semantics for Topic-Oriented Multi-Document Automatic Text Summarization

TITRE DE LA THÈSE / TITLE OF THESIS

Diana Inkpen

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Stan Szpakowicz

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Nathalie Japkowicz

Dwight Deugo

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Shallow Semantics for Topic-Oriented Multi-Document Automatic Text Summarization

Darren Kipp

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the degree of Master of Computer Science (MCS)

November 2008

Ottawa-Carleton Institute for Computer Science
School of Information Technology and Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-48610-8
Our file Notre référence
ISBN: 978-0-494-48610-8

Abstract

There are presently a number of NLP tools available which can provide semantic information about a sentence. Connexor Machinese Semantics is one of the most elaborate of such tools in terms of the information it provides. It has been hypothesized that semantic analysis of sentences is required in order to make significant improvements in automatic summarization. Elaborate semantic analysis is still not particularly feasible. In this thesis, I will look at what shallow semantic features are available from an off the shelf semantic analysis tool which might improve the responsiveness of a summary. The aim of this work is to use the information made available as an intermediary approach to improving the responsiveness of summaries. While this approach is not likely to perform as well as full semantic analysis, it is considerably easier to achieve and could provide an important stepping stone in the direction of deeper semantic analysis. As a significant portion of this task we develop mechanisms in various programming languages to view, process, and extract relevant information and features from the data.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my supervisors for their support, advice, ideas and suggestions. In particular, I wish to extend thanks to Dr. Diana Inkpen for her comments, constructive criticism, suggestions, ideas and never ending enthusiasm. I want to thank Dr. Stan Szpakowicz for providing insights from his extensive experience and knowledge of the subject. Most importantly, thank you both for injecting touches of reality into this process when it was desperately needed.

I would like to extend special thanks to Terry Copeck for assisting me with obtaining the data from past DUC competitions as well as providing me with the various data models that he and others have developed over many years.

I would like to thank my summary evaluators who performed evaluation work for me without any sort of financial compensation. They were Terry Copeck, Amanda Droeske, Diana Inkpen, Alistair Kennedy, Martin Kipp, Martin Scaiano and Stan Szpakowicz

I would to thank my lab mates and other members of our research group. They were all excellent people to work around and always seemed to be available when I needed a favour.

I want to extend a special thanks to my parents for all you did to assist and support through all of my education.

Chapter 1

Introduction

The increasing amount of written information which is now available makes it considerably more difficult to find relevant information in an efficient manner. For this reason, it has become necessary to utilize a variety of language tools for searching for documents. While search engines can retrieve entire documents which may contain relevant sections, they are generally less effective at finding the specific portions of documents which contain the relevant facts and other information required by the user.

Ideally, the user would have available tools which can retrieve only the information relevant to their topic and condense it into a readable form which is also limited in length so that it can be read quickly. This problem, creating a topic-oriented summary, is a difficult natural language processing task. The problem becomes even more difficult as the set of source documents grows.

1.1 Summarization

According to Mani (2001), "the goal of summarization is to take an information source, extract content from it, and present the most important content to the user in a condensed form and in a manner sensitive to the user's or application's needs." The other obvious requirement for automatic summarization is to perform this task automatically.

Within computer science, automatic summarization falls under natural language processing (NLP), an area of artificial intelligence. As is common in other NLP problems some useful ideas and concepts can be found within linguistics. For automatic summarization, the

professional field of human written summarization (also known as abstract writing), is certainly a source of information and ideas.

1.2 Definitions of Types of Automatic Summarization

In a broad sense, the idea of summarization is to produce a condensed form of an information source that is geared to a particular need or set of needs. As a consequence of the breadth of this topic, the task of summarization has many different parameters, creating a wide variety of sub-tasks. These parameters have been defined and added to over time. These are described in Mani (1999), Mani and Maybury (1999), Spärck Jones (1999), Hovy (2001), and the Document Understanding Competition (DUC) (2007b). There are presently a total of nine such parameters. They can be thought of as axes on which to define any summarization problem. We discuss them in the following subsections.

1.2.1 Compression Rate

Compression rate is defined as the summary length divided by the source document length. Several off-the-shelf summarization systems, such as the one embedded in Microsoft Word, offer this as a parameter when the user is creating a summary. Some summarization systems do not work with a particular compression rate but rather an absolute maximum size of the desired summary. In such cases the compression rate will depend on the size of the source documents. The above-mentioned system within Microsoft Word also offers this option.

1.2.2 Audience

The audience parameter specifies what type of information the summary is going to provide. The two broad categories are generic and user-oriented. Generic summaries summarize the entire document. User-oriented summaries allow the user to specify some form of question, prompt or information request. It is also possible to have user-oriented summaries that respond to a standing information request rather than providing a query each time. This could, for example, be used to provide a regular update on scientific discoveries in a particular field.

1.2.3 Abstract or Extract

This parameter is focused on how the summarization is performed. The two types are abstracts or extracts. Extract summaries are produced using segments of the existing documents. These segments can be sentences, paragraphs, phrases, or even individual clauses. Abstracts, on the other hand, are produced by forming new sentences or generalizations on existing sentences. At the current time, almost all automatic systems use extracts while a human writing a summary would almost always write an abstract.

1.2.4 Purpose

Summaries can have one of several purposes. They can be informative, indicative, or critical. An informative summary would be meant to replace the original document. They are essentially an abbreviated version of the original document. An indicative summary provides less information than an informative summary. It would only suggest what the document is about without giving away all of its content (Kan 2001). Summaries with this purpose would generally be best suited to media used for amusement. A few examples of indicative summary include book jackets and movie trailers. Critical summaries are not as likely to be found generated automatically because they provide additional content which does not originate from the original documents. This content would be in the form of some sort of commentary or criticism of the original content.

1.2.5 Span

The span of a summary specifies the information source and more particularly the size of the information source. In the case of summarizing written documents, either a single document or multiple documents can be used as a source. This is often referred to as single and multi-document summarization. Other units of text such as chapters would also be possible.

1.2.6 Language

Language is another possible parameter of text summarization. While typically the data for a summary is written in a single language, it is also possible that the information comes from several different languages. In such a case, a summary produced with multi-lingual data could itself be multi-lingual or all materials could be translated into one target language and then the

summary be produced in that language. The final possible combination would be to summarize information and then translate it into a completely new language.

1.2.7 Genre

Genre is an important parameter of summarization. Often the genre of source documents is useful in determining what information is most useful for a summary. For example, in a news article, journalists are trained to put the most important information first, the additional information second and non-critical related information last. This is done to ensure that if the news article needs to be trimmed to fit print space, the most important information is certain to remain and be printed. For this reason, many summarization systems which use news articles as a source favor information from beginning of documents and virtually ignore information near the end. In a source document like a research paper, sections further into the paper, such as the results or conclusion are usually far more important than related work or the introduction.

1.2.8 Media

Media is parameter of summarization that broadens the area to beyond simple text documents. While a great deal of work has gone into summarizing text documents, other media types can be summarized. These include audio recordings, tables of data, and even pictures or graphics. In media such as audio recordings either an audio summary could be produced or the audio could be transcribed to text and a text summary produced from the transcript.

1.2.9 Static or Dynamic

The newest parameter of summarization is whether the summary is static or dynamic. Static summaries are produced once and then become a document in their own right. The idea of dynamic summaries was put forth in DUC (2007b). Information available for most topics is constantly evolving. As new information gets added, researched, or discovered there is a need to update summarization versions of the information to reflect changes. This new information could confirm, contradict or expand existing facts on the subject. In all cases, there is a need to update the existing summaries to better reflect the current state of the art.

1.3 Hypothesis/Goal

The goal of this work is to use semantic information to improve responsiveness in automatic multi-document text summarization while keeping other evaluation metrics at approximately the same levels as they have for a system that does not utilize semantic information. By responsiveness we mean the quantity and quality of correct information included in a summary that answers an information need. This only includes the information itself and factors such as the quality of writing. See Appendix A for the responsiveness evaluation instructions.

The general idea of extractive summarization dates back to the 1960s. In that early paper Edmundson (1969) suggested that it will be necessary to "take into account syntactic and semantic characteristics of the language and the text", rather than simply utilize gross statistical evidence. To date, full-blown semantic analysis is still evasive. It is however possible to use a hybrid style of design and utilize some semantic information within a statistical structure.

Furthermore, based on several studies discussed in the literature review (see chapter 2), it appears that there still remains some potential to improve summaries while keeping with extractive summarization methods. On that basis, we hypothesize that using some shallow semantic information can improve responsiveness of extractive summaries. This will be done by producing features using a semantic parser/analyzer. Such features can then be added to any extractive summarization system which uses feature vector architecture. These features can take a variety of forms. They can be an additional weight added to a sentence ranking function or rules designed to completely filter out sentences.

For this work we will use Connexor Machinese Semantics (Connexor 2003a), a semantic analysis tool, which provides an extensive amount of information about sentences. The first challenge is developing an efficient means of extracting useful information and features from the XML-formatted parses provided by the tool. The second task is to conduct a detailed examination of what features and information the analyzer provides and determine how it might be useful for the task of automatic summarization.

Once some features have been chosen, we will perform frequency analysis on the document set, the model summaries and the peer summaries from the 2005 DUC competition

(DUC 2005). This frequency analysis will count the number of occurrences of particular features in an attempt to determine if certain features are more likely to appear within summaries. Using this information a final collection of features will be chosen.

Finally, a series of experiments will be conducted in an attempt to utilize these features to improve the responsiveness in topic-oriented multi-document automatic text summarization. These summaries will be automatically evaluated using a reverse-engineered summary content unit (SCU) corpus (Copeck and Szpakowicz. 2004). A selection of the topics from the top-performing system configuration will then be manually evaluated by volunteer evaluators to provide a comparison with systems presented at DUC. In order to make comparison easy, we will define our summarization task identically to that of the summarization task at DUC (2005). We produce 250-word summaries from collections of 30 to 50 documents. The summary will be based on a topic statement of one to four sentences.

1.4 This Thesis

This thesis contains 5 chapters. Chapter 1, the introduction, introduces some of the basic concepts of automatic summarization, along with some key definitions. It also discusses the problem this thesis is addressing and goals of this work. Chapter 2 provides a review of some literature relevant to automatic text summarization in general, as well as the more specific problem of topic-oriented multi-document text summarization. This includes a review of current evaluation techniques. Chapter 3 describes the Machinese Semantics tool, the features extracted from it and what features were used in the production of summaries. It also describes the experiments conducted using machine learning and heuristics to produce summaries. A summarization system, which was developed as part of this work, is also described. Chapter 4 provides the results of automatic evaluation and the manual evaluation of the summaries produced by a top-performing heuristic. Chapter 5 provides the conclusions and a discussion of a number of possible directions for future work using a semantic analyzer such as the one by Connexor.

1.5 Contributions of this Thesis

In this thesis I demonstrate how to utilize some information provided by Connexor Machinese Semantics to improve responsiveness in automatic multi-document text summarization. A variety of simple and complex features are extracted from the information that Connexor provides. These features are shown to produce results which are comparable with other systems performing the identical task. I demonstrate a possible way to utilize this information within a common configuration of text summarization system.

I compared a variety of automatic and manual evaluation methods on summaries produced by a system that uses the Connexor features and by other systems that do not. This includes a simple lexical matching system to provide a baseline measure for summarization systems.

Finally, I created tools for the examination and utilization of information provided by Connexor Machinese Semantics. An XSL style sheet is provided in order to allow viewing the information from Connexor Machinese Semantics as nested feature structures within a web browser or another HTML-rendering tool. I also provide a series of Java class structures for storing and processing information contained within each feature available.

Chapter 2

Related Work

There is a considerable range of automatic text summarizations problems, tasks, approaches and methods. Spärck Jones (2007) points out that over the last decade there has been a surge of interest in automatic summarizing. There have been a number of workshops along with an annual friendly competition at the annual Document Understand Conference (DUC) which we will describe later.

2.1 Automatic Summary Production

There have been many possible rules, heuristics, processes and schemes created to produce automatic summaries of various media. Many of these processes have been borrowed, at least in part, from other areas of natural language processing and related areas. This includes question answering, information extraction, and machine leaning. Despite having such a variety of methods, many automatic text summarization systems function similarly by using extractive summarization. This process involves choosing certain sentences from the source text which, by some measure, appear to be the most relevant. The only considerable difference is the process is the method of selecting which sentences should be extracted. Some form of clean-up can be added to the end of summary creation process to address readability. Readability is particularly important in extractive summarization since sentences inserted into a paragraph may not flow well together. There can also be issues such as dangling or ambiguous pronouns which can further make the summary difficult to read and understand

2.1.1 Extractive Summarization

Statistical approaches to various types of automatic summarization have been considered since at least 1958 in a work done by Luhn (1958). Luhn looked at selecting key words and then choosing sentences of most significance based on the presence of the key words. This task was performed by counting word frequencies. Common words, later termed stop words, were removed from the list.

"*New Methods of Automatic Extracting*" (Edmundson, 1969) was another early work on producing summaries based on sentence extraction. The summarization problem studied by Edmundson was generic single document summarization. The paper looked at the performance of a sentence extraction algorithm using factors such as cue phrases, key words, titles, and location within a document.

The central methodology was to give the various factors different weights and ultimately compute a score for each sentence in the document. Much of the paper's assessment includes measures which are either no longer relevant or of minimal relevance such as keypunching costs. One statement does however stick out: "It is now beyond question that future automatic abstracting methods must take into account syntactic and semantic characteristics of the language and the text: they cannot rely simply upon gross statistical evidence".

Although 39 years have passed since Edmundson's paper, most systems still use a similar framework for summarization. At the 2006 DUC conference all systems utilized some form of extractive summarization. The major differences were the features used and the ranking algorithms.

Mani (2001) discusses several ideas about automatic summarization. A notable one is that humans write abstracts and not extracts. Automatically producing an abstract would require natural language generation (NLG). At the current state of the art, NLG technology is not sufficient developed for producing text about general topics using a variety of language constructions. As a result, and as demonstrated by past DUC conference tasks (DUC, 2005; DUC, 2006a; DUC 2007a), all current automatic summarization is based on extraction. Other work on generic summarization would likely demonstrate the same thing.

Extraction involves building summaries using information cut and pasted from the original source documents. While there is no fixed rule regarding the size of the information extracted as a unit, individual sentences are often used. It is possible to select smaller units such as phrases or larger units such as a group of sentences or paragraphs. On reason sentences tend to be chosen is that they are the smallest fully grammatically correct unit that can be selected. Choosing smaller units requires additional work in order to create a coherent text output. If larger units such as paragraphs are selected it becomes harder to included a variety of information in a summary due to the space requirements of the larger units.

In Mani (2001), it is noted that most extractive summarization utilizes shallow analysis and rarely any semantic information or analysis. Extraction is also less suited to summarization with a large compression ratio. Multi-document summarization by its nature often has very large amounts of information as input, and as a consequence, very large compression ratios. For these reasons, Mani believes that extracts are not sufficient for multi-document summarization. Counter to this assertion, is the past DUC competitions involving multi-document text summarization. As detailed later, all systems at these competitions utilized extractive summarization.

To add to Mani's argument, other work comparing summaries produced by humans with summaries completed by machine found the two somewhat incompatible. One such study, Copeck and Szpakowicz (2004), examined the differences in language usage between the source documents and summaries produced manually. In particular, they found a very low overlap between the vocabulary usage in the human summaries and the source document set. The study used a variety of summary lengths and both phrase matching and token matching. They also added other processes such as stemming to improve the matching. In terms of a token matching on a 200 word summary, they found only a 53% agreement in vocabulary between the documents and the summaries. The results were even lower when comparing summaries to each other: 22%. This leads to the conclusion that, when writing a summary, humans will use their own wording and terminology to describe things rather than the terminology in the original documents.

It is easy to infer some reasons why this might be. One reason may be the efficiency of terminology. Utilizing outside language in the summary may allow the person writing the

summary to pack more information into the summary by using more efficient terminology. A single term may, for example, replace a larger description. A second practical reason may be that it is easier to produce fluent summaries when writing sentences rather than extracting them. When sentences are extracted to produce a summary, they may end up being somewhat rammed together producing a result where adjacent sentences may not fit well together or may be hard to read together. A final reason for the low vocabulary overlap may simply be that most people prefer to vary their writing style to increase readability and perhaps decrease the reader's boredom or fatigue.

Despite Mani's assertion that extraction is insufficient for multi-document summarization, it is difficult to see how anything else would be possible at this point in time given the current state of the art in automatic summarization and a number of other areas of natural language processing: natural language generation and natural language understanding. Mani does also note that most analysis is usually shallow and rarely semantic. Full semantic analysis applied to this task is also unlikely to be feasible until further into the future. It may be possible to utilize some shallow semantic information within the present extractive framework to improve the responsiveness of the resulting summaries.

Although extraction is not sufficient for multi-document summarization, another study does demonstrate that some improvements over the present state of the art are still possible without moving beyond extraction. Lin and Hovy (2003) made an attempt to define an upper bound for the performance of extractive summarization their paper "*The Potential and Limitations of Automatic Sentence Extraction for Summarization*". They drew several conclusions based on unigram co-occurrence scores. The experiment produced all possible summaries of a given length. They then selected the best possible summary based on unigram co-occurrence scores with the model summaries. The first major conclusion was that, because summarization is loosely defined, there is significant variability in inter-human agreement on summarization topics. They determined the difference between the maximum agreement and the minimum agreement within DUC 2001 tasks to be 18%. State of the art systems performed about 10% below the performance of an average human. They also determined a relationship with the size of a summary and performance. There was a considerable improvement (15%) in scores for 150-word summaries compared with 100-word summaries. Computing average

unigram co-occurrence score for the full text produced a score 0.883, a score 24% higher than the 150-word summaries.

Teufel and Moens (1997) looked at sentence extraction as a classification task. They attempted to select sentences for a summary based on their meaningfulness. In order to perform this as a classification task, sentences are classified as meaningful or not. No degree of meaningfulness was specified. They then used cue phrases, sentence length, sentence location, thematic words and titles. When evaluating their methods, they refer to "precision and recall" as a single value. It is not clear whether they mean f-measure or they are using some other means of combining precision and recall. Their methods were individually able to achieve a precision and recall of between about 21% and 55%. Cumulatively, they achieved a precision and recall of 68.4% against a baseline of 28%. This shows some potential for classifying sentences based on their use within a summary. The results may not be good enough to use only this method of summarization. It also suffers from the problem of returning more sentences than will fit in a summary. Consequently a secondary process is required to in order to choose which sentences make the final summary.

2.1.2 Topic-Oriented Multi-document Summarization

Mani (2001) states that the goal of multi-document summarization is:

> "... to take an information source that is a collection of related documents and extract content from it, while removing redundancy and taking into account similarities and differences in information content and present the most important content to the user in a condensed form and in a manner sensitive to the user's or application's needs"

Eliminating redundancy is an important goal of multi-document summarization. Many current systems are either not advanced enough to remove redundancy or rely on details such as the source texts being sufficiently different to provide non-redundancy. Some systems do however include algorithms specifically designed to eliminate redundancy.

Topic-oriented multi-document summarization adds one more element to this, the topic. The topic is generally specified as some form of title, sentence or group of sentences describing what information is to be included in the summary. Ideally, this information would be understood by the summarization. In general, more shallow methods such as text matching are

used here. The main complication is that, in addition to the high compression ratio for multi-document summarization, we must also orient the summary to the information requested in the topic.

2.1.3 Comparison with the Work of Human Abstractors

An obvious place to search for techniques for any automated task is to look at how humans would perform the task manually. There have been a few studies into the methods professional summarizers use to produce abstracts. Three such works were Cremmins (1996), Pinto Molina (1995), and Niggemeyer (1998). All three suggest that there are multiple steps to the process which follow an approximate flow starting with an initial reading, followed by selecting relevant facts, followed by some re-organization and ultimately the production of the summary. It is also notable that although there are likely common steps and mechanisms for performing a task, it is also probable that a group of humans would differ in how they perform a given task. This is similar to the way that these three processes described in the three studies have slight differences between them.

In all of the above cases, the abstracts tend to be oriented towards generic abstracts and probably are done on single documents only. It is however not inconceivable to extend these basic ideas to multi-document and user-oriented abstracts.

In the case of multiple documents, the only major differences are that there is more information input and that some information may potentially be redundant or contradictory between different documents. The first difference is handled in the first stage. It simply means that there is more information to absorb and process. The second difference would be handled by the organizational and summary production stage. This stage would have the added task of removing redundant information. Neither of these differences would likely cause any material differences in the way in which a professional abstractor is likely to perform his or her work. Contradictory information on the other hand, is a considerably harder issue to handle. This is because information can often change over time. It is not an insignificant issue to simply identify information that does not match. This problem is left unsolved at this point.

Extending this to a user-oriented abstract would require further revisions to the process. The initial information collection stage would be altered considerably. With a generic abstract,

the document would need to be read to determine what the author's keys points are. In the case of a user-oriented summary or abstract, the reading stage is looking for information relevant to the user's needs and the author's points are less relevant. The organization and summary production stages also differ in user oriented versus generic summarization since information needs to be presented according the user's needs rather than the manner in which the original author presented information.

2.2 Summary Evaluation

Harman and Over (2004) conducted a study on how differences in model summaries affect evaluations of summaries. The study found considerable variation in the vocabulary usage with model summaries. This difference was empirically attributed to differences in what details are included within a summary. This follows logically from the fact that summarization is a loosely defined task and that there is no clear and agreed definition of precisely what should be included within a summary.

The study concluded that, on average, variability in the model summaries has minimal effect on the overall evaluation across the entire collection of document sets. The study does note that even if there is no noticeable effect on the average, there can still be an effect on individual document sets for a certain summary topic.

There are both intrinsic and extrinsic evaluations of summaries (Spärck Jones and Galliers, 1996). Intrinsic evaluation is based on the content of the summary while extrinsic evaluation is based on how it affects a particular task. In generic summarization the latter is not useful since there is no defined use for the summaries that are produced.

Santos et al (2004) proposed to automatically evaluate summaries using a graph-based method. Document graphs were produced using the Link parser (Sleator and Temperley, 1993), and the noun phrases were extracted. The final graphs were produced using a series of heuristics. The similarity of the resulting graphs was then compared. The paper only notes that different measures produce different evaluation rankings, but does not attempt to position any particular method as being superior.

A proposal for task-based evaluation of text summarization systems, Hand (1997) looks at an extrinsic evaluation. It looks at the suitability of a summarization system for categorization, and ad hoc retrieval tasks. This is another means of evaluating a summarization system which is not suitable for summaries produced for more generic purposes.

2.2.1 Summary Content Units / Pyramid Evaluation

Pyramid Evaluation is a form of manual evaluation of summaries (Harnly et al., 2005). It looks only at the content of summaries and not at such phenomena as grammaticality, or style. It is based on comparing the content of the summary with the content of a set of human-written model summaries. These model summaries are considered the gold standard for summaries on a given topic using a given document set. Such model summaries are used in a variety of summary evaluation techniques. This form of evaluation was utilized at the Document Understand Conferences (DUC), which is described in detail later.

Model summaries may not utilize the same language as machine-generated summaries. One of the major reasons for this is that the summaries produced by humans do not employ the same phrases and language as the original documents (Copeck and Szpakowicz, 2004). Because of this, it is not possible to compare summary content word for word with what is in the model summaries. Instead the comparison is based on what is said and what ideas are expressed, rather than what words are used. Figure 2.1 provides and example of two sentences expressing similar ideas:

1) John accepted a new job at Software Incorporated.
2) Software Incorporated hired John as a new employee.

Figure 2.1: Two Sentences Expressing Approximately the Same Ideas

Summary content units (SCU) are small phrases or snippets found within sentences which are relevant to the topic or question which the summary is based on. These units are created manually from the facts and ideas that appear in human-written model summaries for the topic. The SCUs can then be manually matched with facts and ideas expressed in automated summaries. From this matching a score is produced. A tool is used to keep track of all the information (Nenkova and Passonneau (2005). The tool is also capable of removing redundant work such as scoring the same sentence appearing in more than one summary.

SCUs can have different values associated with them. Certain facts are considered more important than others. Given that in summarization space is limited and therefore the number of facts that can be expressed is limited as well, stating the most important facts has great importance. In pyramid evaluation, this importance is determined by how many of the model summaries the summary content unit appears in. Essentially, a fact stated in more model summaries is considered to be worth more. The maximum possible score or weight that a summary content unit can have is determined by the number of human written model summaries for each topic. In the 2005 DUC competition, 7 model summaries were used for each topic, while in the 2006 DUC competition 4 model summaries were used.

For each topic that a pyramid evaluation was completed for, the scores are scaled differently. This is due to the fact that, for different topics, the amount of information available which answers the query varies. This is primarily because of the differences in the amount of useful information available in the source documents. To correct for this difference, the pyramid scores are converted into mean modified pyramid scores. This allows the performance of summarization systems to be compared and averaged across multiple topics. Mean modified pyramid scores are produced by dividing the total SCU score by the average pyramid score of the model summaries for each given topic. The dividing factor is different for each topic. The resulting pyramid score can be thought of a percentage of the average weighted SCU count achieved by the model summaries.

Completing pyramid evaluation is largely a manual process. At recent DUC competitions, the pyramid evaluation was is limited to the submissions of the groups who agreed to help with the pyramid evaluation. The task involved two steps. The first is to produce the pyramids from the model summaries and the second step is to identify summary content units within each of the submitted summaries.

While the initial completion of SCU evaluation for a topic is completely manual, it has been shown that it is largely possible to reverse-engineer the pyramid evaluation to associate SCUs and their weights with sentences in the source documents. This technique is based largely on the fact that most teams participating in the DUC competition either selected complete sentences or utilize a sentence selection techniques where sentences may be slightly altered or trimmed after they have been selected. Such reverse mapping linked sentences from summaries

to sentences within the source documents in 2005 and 2006. Using data from year 2005 DUC task 83% of sentences appearing were linked to sentences in the source documents. Using data from the 2006 DUC task, 96% of the sentences in summaries were linked to source document sentences. (Copeck et al., 2006).

Such a reverse mapping allows extractive summarization systems to be trained and tweaked to a pyramid score without the need for human labour. It does have a few small caveats. The set of sentences in the source documents are placed into three categories:

1) Sentence which were selected by at least one submission to DUC that contained at least one SCU
2) Sentence which were selected by at least one submission to DUC that did not contain any SCUs
3) Sentences which were not selected by any system submitted to DUC

It is therefore not possible to know if a sentence that was not selected by any system at DUC contains any SCUs. As a consequence to this, it is not possible to do a complete pyramid evaluation using the reverse-mapped SCU corpus. This is however counter-balanced by a few points. The first is that given the number of submissions to DUC, sentences in the third category are significantly less likely to be selected by a system since none of the submitted systems selected them. Furthermore, many of the sentences in that category do not contain any key words found in the query, meaning that virtually no query-based summarization system will select them. The second counter-point is that the only other means of doing automated evaluation of summaries, the ROUGE system (see the next section), is based on producing scores which correlate well with summaries which scored well when evaluated with manual measures. Since correlations are by definition uni-directional, this does not make for a strong measure to tune a system to.

The SCU corpus is not presently available for all DUC topics due to the levels of participation by groups who participated in the DUC competition. This does limit the amount of data available for tuning or training systems with this measure. This will also reduce the performance of any tuning in much the same way we would expect when having a machine learning system use less training data. It otherwise does not have any effect on the use of this data.

2.2.2 The ROUGE System

The ROUGE system is a completely automated system for evaluating summaries. ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. (Lin, C. 2005) It evaluates summaries based on a set of statistical measures which compare a candidate summary to set of human-produced model summaries. There is a set of 4 measures, although not all have been used for evaluation of summaries at DUC (Lin, 2005). These measures all count words in various different ways. In the remainder of this section there is a detailed overview of the measures which had been used at DUC 2005. The most important thing to note when looking at the details of each of these measures is that there is no direct examination of content.

The first such measure is ROUGE-N, which measures the N-gram Co-Occurrence Statistics. This measure is the total number of n-gram matches between a given summary and the reference (model) summary divided by the total number of n-grams in the reference summary. If there are multiple reference summaries, as is the case with DUC, a ROUGE-N score is computed pair-wise with each reference summary and the maximum score is kept.

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

Where n is the length of the n-gram, $Count(gram_n)$ is the maximum number of n-grams appearing in the reference summaries, and $Count_{match}(gram_n)$ is the maximum number of n-gram matches (co-occurrences) between the reference summaries and the candidate summaries.

The second ROUGE measure is ROUGE-L, which measure the longest common subsequence. A longest common subsequence between two sentences is the longest ordered list of words that is common between them. This measure is computed as an F-measure of the longest common subsequence between each pair of sentences (one sentence from the reference summary and one sentence from the candidate summary). The precision is computed by taking the longest common subsequence between the sentences in the model summary and the sentences in the candidate summary and dividing by the length of the candidate summary. The recall is calculated in the same fashion except that the divisor is the length of model summary. An F-measure is then computed.

$$R_{lcs} = \frac{LCS(X,Y)}{m}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n}$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}}$$

Where X and Y are summaries of lengths m and n, LCS(X,Y) is the longest common subsequence between them, and β is a parameter (as mentioned previously, 8 is used at DUC).

At the summary level, the measure is computed similarly except that the precision and recall are computed using the sum of the union longest common subsequence measures between each reference summary sentence and each candidate sentence. The F-measure is computed in a usual fashion.

$$R_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{m}$$

$$P_{lcs} = \frac{\sum_{i=1}^{u} LCS_{\cup}(r_i, C)}{n}$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}}$$

Where r_i is a sentence from a reference summary, C is the set of sentences in a candidate summary, β is a parameter (8 is used at DUC), and $LCS_{\cup}(r_i, C)$ is the union longest common subsequence between a reference summary sentence and each sentence in the candidate summary. The union longest common subsequence is the union of the longest common subsequences between each pair. This produces an ordered list of matching words between the two sentences.

The third ROUGE measure is ROUGE-W. This is a weighted longest common subsequence measure (WLCS). The is measure gives more value to common subsequences that are consecutive matches rather than common sequences split up by other words which are not part of the sequence. WLCS is computed using a dynamic programming algorithm. A weighting function is supplied as a parameter. The only requirement is that it has the property that consecutive matches receive a higher score than non-consecutive matches or in mathematical terms $f(x+y) > f(x) + f(y)$.

The paper on ROUGE also suggests that in order to normalize the ROUGE score a function with an inverse of similar form is also preferable. An example of this would be $f(k) = k^2$ where k is the number of consecutive matches up to a current position in the dynamic programming table. The last value in the dynamic programming table is the eventual ROUGE W score.

The score is computed as follows:

$$R_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(m)}\right)$$

$$P_{wlcs} = f^{-1}\left(\frac{WLCS(X,Y)}{f(n)}\right)$$

$$F_{wlcs} = \frac{(1+\beta^2)R_{wlcs}P_{wlcs}}{R_{wlcs} + \beta^2 P_{wlcs}}$$

Where X and Y are sentences of lengths m and n, f^{-1} is the inverse of function f.

The fourth ROUGE measure is ROUGE-S: Skip-Bigram Co-Occurrence statistics. Skip-bigrams are ordered pairs of words from a sentence that can have an arbitrary gap between them in the original sentence. The ROUGE-S score is calculated using a simple F measure of the number of overlapping skip-bigrams between the candidate summary and the model summary.

$$R_{skip2} = \frac{SKIP2(X,Y)}{C(m,2)}$$

$$P_{skip2} = \frac{SKIP2(X,Y)}{C(n,2)}$$

$$F_{skip2} = \frac{(1+\beta^2)R_{skip2}P_{skip2}}{R_{skip2} + \beta^2 P_{skip2}}$$

Where SKIP2(X,Y) is the number of skip bigram matches between a candidate summary of length m and a reference summary of length n. C is the standard combination function.

ROUGE-S has a few important limitations. The first is that combinations of things such as stop words and other words of limited importance due to frequency could alter the score in an unfavorable fashion. To help prevent this, the authors suggest limiting the distance between words in the skip-bigrams.

A ROUGE-S score limitation is discovered by considering an example of two partial sentences from Lin (2005).

1) police killed the gunman
2) gunman the killed police

Without getting caught up in whether these particular partial sentences make much sense, it is worth noting that attempting to match these two results in a ROUGE-S score of zero. The reason for this is that they have absolutely no skip-bigrams in common. Conversely a simple word matching score would give these two a perfect score. To solve this problem, the authors suggest adding a beginning of sentence marker at the start of each sentence. This marker can be a member of the skip-bigrams. This is called the ROUGE-SU measure.

Various configurations of these measures we studied by calculating correlations with human evaluation measures using data from DUC 2001, 2002, and 2003 (Lin, 2004). For evaluations at DUC 2005 two ROUGE measures were computed: ROUGE-2 (ROUGE-N with bigram matching) and ROUGE-SU4 (ROUGE-SU with a skip distance limit of 4). At later DUC

competitions experiments were done with an additional measure which utilized word dependencies from the Minipar parsing system (Lin 1999).

The ROUGE system overcomes some of the limitations of the SCU evaluation. Since it is entirely automated it can be run on any topic that has sample summaries. It is also easy to compare summaries with this measure. The ROUGE system does however have a few limitations of its own. It is based on a study showing that its scores correlate well with summaries that score well in human evaluation. This, unfortunately, makes it unsuitable to use for tuning a system. Nonetheless, we can use it to evaluate a system tuned by some other means. Sjöbergh (2007) demonstrates summaries that would produce high ROUGE scores that would not be considered good summaries by a human reader. This is a general limitation of any automatic statistical evaluation method.

2.2.3 Manual Evaluation

Manual evaluation involves a judge or team of judges reading each topic and summary and applying some form of score or mark to each. There is certainly no single way of doing this. There are also a number of possible aspects of a summary that could be evaluated. These range from the content of the summary to the overall readability and grammar. In order to be able to compare evaluations between summaries, an established evaluation scheme is required. For this we look towards the DUC competitions.

At the DUC competition, summaries were evaluated manually using a variety of measures (DUC 2006). Throughout the years of the DUC competition these measures stayed generally the same with only a few minor changes. At the 2006 competition, summaries were evaluated manually using five measures of linguistic quality/readability and two measures of summary content.

The five measures of linguistic quality were grammaticality, non-redundancy, referential clarity, focus, and structure and coherence. The two measures of content were responsiveness and overall responsiveness. Responsiveness measured the information the summary presented. It was measured for each summary individually. Overall responsiveness applied a single score to an entire set of summaries, spanning all topics, which were produced by a single system. In producing the overall responsiveness score, the evaluators were not given access to what

responsiveness scores they had assigned individual summaries. Instead they were to produce a score based on their overall feeling about the set of summaries. All linguistic quality and content measures are based on a five-point scale.

1. Very Poor
2. Poor
3. Barely Acceptable
4. Good
5. Very Good

The grammaticality measure was to ensure that the summary had no:

- Datelines
- System-internal formatting
- Capitalization errors
- Obviously ungrammatical sentences

The non-redundancy measure took into account all forms of redundancy including repeated sentences, repeated facts and repeated use of proper nouns when pronouns could have been used. In multi-document summarization, duplication is a non-trivial problem, particularly when the document source is news articles. Some newspapers produce articles by modifying articles from wire services such as the Associated Press or Reuters. As a result, articles in different newspapers can contain similar sentences and similar overall structure. The other source of redundancy in multi-document summarization of news is that different articles will need to introduce things like job titles to their readers. Extracting such sentences from several articles can lead to a person's job titles being mentioned numerous times.

The referential clarity measure ensures that it is clear exactly what pronouns and noun phrases refer to. More specifically, within this measure there would be a deduction in score for orphan pronouns, or situations where it is not clear what a reference is referring to. Summaries produced using extractive methods of automatic summarization can very easily contain stray anaphora if care is not taken to either replace these or avoid selecting these sentences.

The focus measure checks to see that sentences contain only information which is related to other information in the summary. In extractive summarization, it is conceivable for a sentence from another source to contain clauses discussing matters unrelated to the topic even though elements like named entities match the topic. An example of this would be discussions between heads of state or heads of government from a pair of countries. They are very likely to discuss a range of matters. Consequently, a sentence matching both leaders' names still has only a limited chance of turning up useful information.

The structure and coherence measure examines whether the summary is well-structured and well-organized. Automatic summaries can easily become a heap of information presented in an arbitrary way. Such a structure would be penalized in this measure. Instead, it is desirable to have information organized in a logical way, such a by events or along a timeline.

On the content side, responsiveness in DUC was measured in two ways. The first measure was done on a topic by topic basis and then a mean is computed to produce a score. The second measure was identical to the first except that it was done across the entire collection of topics in a single score, rather than producing a separate score for each topic separately and then averaging. Both responsiveness measures look at how well the information requirement in the topic statement was met.

In the first responsiveness measure the human evaluator is asked to consider the job the system has done for a certain topic only. In other words a very poor system could, for whatever reason, perform very well on a certain topic. In such a case it would achieve a good score on that topic and presumably a poor score on all or most of the others.

In the second responsiveness measure the evaluators were asked to consider the responsiveness of a system across the entire spectrum of topics. Further to this requirement, the evaluators were asked not to consider the scores given on individual topics nor were they given access to the scores they had given to individual topics.

2.3 Document Understanding Conference

The Document Understanding Conference (DUC) is an annual workshop on Automatic Text Summarization that is both funded and operated by the National Institute of Science and

Technology (NIST) in the United States. Starting in 2008, this event will now be called the Text Analysis Conference (TAC) (TAC 2008). Text summarization will be one of a few tracks at it. A central feature of the conferences is one or more summarization tasks. To some degree, these tasks take the form of friendly competition, although no winner is formally declared. In recent years, there have been about 30 submissions to the main task. These submissions come from a variety of groups at universities, research institutions, and private companies around the world. Although the largest number of submissions comes from within the United States of America, the remainder is fairly geographically varied. The first edition of the conference was in the year 2001 and a new edition has been held roughly each year since. The conference provides several important things to the summarization community at large.

A first significant contribution of DUC to automatic summarization research is the fact that it organizes the summarization community around current and relevant problems. With numerous systems participating in an identical task, it is very easy compare a varied assortment of works and approaches to the problem. The scoring tables provided by the organizers serve as an excellent means of determining the state of the art performance for a given problem. This differs from standard academic conferences in natural language processing, where even papers within a particular track often deal with disjoint problems.

2.3.1 Document Understanding Conference Research Tasks

The DUC conference has run a varied range of system tasks throughout its lifespan. Many of the tasks have involved single- or multiple-document summaries that were either generic or user-oriented. These summaries have typically been approximately 100-250 words in length. The conference has also tried a variety of other tasks including very short headline type summaries, cross-language summaries, and summaries of varied focus. The conference will often run a particular task for several years. This allows further development in that particular area.

The second major contribution of DUC to the summarization community is data. Each year NIST provides test data designed for the current task. NIST also, on some occasions, has provided a limited amount of sample data. This data can be used for the competition task and also for subsequent research on summarization. In general, the data for a specific task is fairly varied. Many of the main DUC tasks are run for several years allowing for a fairly large

collection of training and testing data to be amassed. It is also common for NIST to provide several manually-written model summaries for each topic. These model summaries assist in both evaluation and subsequent analysis and discovery.

2001	Single and multiple document summaries
2002	Single and multiple document summaries
2003	Very short summaries (10 words); Short summaries focused by events; Short summaries focused by viewpoints; Short summaries in response to a question
2004	Very short single-document summaries; Short multi-document summaries focused by events; Very short cross-lingual single-document summaries, Short cross-lingual multi-document summaries focused by events; Short summaries focused by question
2005	User oriented multi-document summaries
2006	User oriented multi-document summaries
2007	User oriented multi-document and update summaries

Figure 2.2: Document Understanding Conference Summarization Tasks

The third contribution is evaluation. The DUC conference provides several evaluations of systems. These evaluations range from human to automatic. Some of this evaluation is completed by employees hired by NIST, some is automatic, and some of the evaluation requires a significant volunteer effort from around the summarization community. Several recently developed summarization evaluation systems have also been used and tested as part of the DUC competition.

In recent years, the evaluation completed by employees of NIST involves a human evaluation in which a series of measures are used. This evaluation was the most comprehensive. NIST also runs a fully automated evaluation of the content of summaries using the ROUGE software. A third evaluation of content is conducted and coordinated by volunteers from a selection of the DUC participants.

For the past three years, DUC has focused the summarization community around a user-oriented multi-document summarization. The main task for 2005, 2006 and 2007 was to produce a 250 word summary from a collection of 25-50 documents. The summary produced is based on a 1-4 sentence topic which details the information requirements.

For the 2008 competition this task is further evolving into update summarization. Update summarization involves altering an existing summary to take into account new information. The current update summarization task is still based on user-oriented summarization.

2.3.2 Using DUC Resources for Research

There are several benefits to matching the summarization task being researched to the summarization task used by the Document Understanding Conference. The main benefit is that the data is designed to be both suitable and fair to that task. For this reason, no modifications to the data need to be made in order to utilize it. Additionally, many existing scripts used for pre-processing the data can also be used.

The next benefit to utilizing the DUC task is that there are a large number of available results from comparable systems. This provides an excellent means of determining the state of the art for a particular summarization problem. Additionally, it provides numerous insights into the differences between systems and the entire spectrum of performance of the different systems.

2.3.3 Review of Methods used at DUC 2006

Among DUC participants, a number of sentence ranking mechanisms were utilized. The first such group uses the same basic weighted feature vector structure Edmundson used. The only difference is in the features. Because the 2006 DUC task involved user-oriented summaries rather than generic ones, the core features generally involve a direct or indirect lexical match. In DUC submissions, numerous additional features have been applied ranging from document location to lexical semantic mechanisms. The second ranking method, used by a few systems, is clustering. The third major method is based on graphs. Within the work done at DUC, we look for systems that utilize semantic information to determine what gaps in methods exist.

The topic statement in topic oriented summarization serves as a more explicit definition of the user's needs, whereas in the generic multi-document summarization case, the user's needs are considerably less clear. While the topic specifies the user's or application's needs, it is still not obvious how to use such information. A number of similar approaches have been attempted at recent DUC but all systems performed some type of lexical matching between the topic statement and the document cluster.

The first approach is lexical matching between the topic statement and sentences, or other units such as phrases or paragraphs within the document collection. Within the DUC task a few different methods of doing this can be found. The first was straight lexical matching. This method was used in the majority of systems. (Zhou et al., 2006; Conroy et al., 2006; Jagarlamudi

et al., 2006; Lacatusu et al., 2006; Vanderwende et al., 2006; Wu et al., 2006; Ye, and Chua, 2006; Fisher and Roark 2006; Li et al., 2006; Schilder and Thomson, 2006; Zajic et al., 2006) Some of the above mentioned systems enhance the lexical matching using techniques such as stemming, lemmatizing, and some form of keyword extraction or stop word removal. Matching can also be restricted to just certain parts of speech (Zhou, Sun, and Lv, 2006).

A second extension on matching between the topic statement and the document set is to weight the matching utilizing the frequency of word occurrence within either the language or the document set. One common way to do this is to utilize TF.IDF (term frequency multiplied by the inverse document frequency) formula to weight the word matches. This was done in a number of systems ay DUC 2006 (Blair-Goldensohn, and McKeown, 2006; Li et al., 2006; Favre et al., 2006; Witte et al., 2006; Zajic et al., 2006; Erkan, 2006). Wu et al. (2006) weighted certain word matches, such as named entities, higher.

Another common improvement on simple lexical matching is to add various lexical resources. The most common such resource was Wordnet (Miller, 1995). Typically, such resources are used to extract synonyms or hypernyms as was done in Melli et al. (2006) and Bosma (2006).

A few systems performed more complex matching techniques. These included tree matching (Schilder and Thomson, 2006), graph matching (Mohamed and Rajasekaran, 2006; Witte et al., 2006) and n-gram matching (Schilder and Thomson, 2006; Fuentes et al., 2006). Other less commonly used techniques attempted at DUC were clustering (Seki et al 2006), lexical chains (Zhou, Sun, and Lv, 2006), and other statistical measures (Alfonseca et al., 2006; Doran et al., 2006).

Although the various forms of lexical matching served as the core measure or mechanism for selecting sentences, a number of other measures or algorithms were used by various systems to enhance the result. Such measures were either added to an Edmunstonian feature vector or were used to pre-filter the document set or post-filter the result.

Sentence trimming is another technique used in summarization. Instead of shortening the information by filtering out entire sentences, it filters out only portions of sentences. This technique is also called sentence shortening or sentence compression. In Vanderwende, et al. (2006), sentences are shortened by first parsing them and then eliminating certain nodes of the

parse trees. Which parts are removed is determined by matching certain syntactic patterns such as noun appositive, gerundive clause, and non-restrictive relative clause.

Three examples given in that work are:

Noun appositive: *One senior, Liz Parker, had slacked off too badly to graduate.*

Gerundive clause: *The kialegees, numbering about 450, are a landless tribe, sharing space in Wetumka, Okla, with the much larger Creek Nation, to whom they are related.*

Nonrestrictive relative clause: *The return to whaling will be a sort of homecoming for the Makah, whose real name _ which cannot be written in English _ means "people who live by the rocks and the sea-gulls."*

In all the examples provided by Vanderwende et al. (2006), a substantial number of words are eliminated. This does however increase the risk of several things. The first is that without careful correction of punctuation and capitalization the sentences can become ungrammatical. There is also an increase risk that good information has been lost in the trimming. Dorr et al, (2003) and Zajic et al. (2004) also describe a trimmer. This was employed by Zajic et al. (2006).

Zajic et al. (2006) use a statistical measure of probability to determine if a sentence is redundant compared to the rest of the summary. Copeck et al. (2006), a system I participated in the development of, utilizes ROUGE scores a similarity measure to compare sentences pair-wise and eliminate one sentence from a pair scoring too high. In both cases, the systems were not formally evaluated and parameters were only tweaked by guesswork.

Among DUC participants we see a broad variety of methods tested. In all systems we see very little of any sort of semantics utilized in ranking sentences. This demonstrates both an area where little work has been done and a possible place to find some improvements in responsiveness,

Chapter 3

Experimental Methodology

3.1 Description of Experiments

A goal of this work is to explore the use of the Connexor semantic parser/analyzer (Connexor, 2003) to improve responsiveness in topic-driven multi-document text summarization. As the literature review discussed by looking at other DUC submissions, very little previous work existed that utilized semantic features in any considerable way. A notable source of datasets for text summarization is the Document Understanding Conference's summarization task. It is impossible to explore using a resource without first determining a training and testing methodology. A topic-driven multi-document summarization task was run at DUC for three consecutive years (2005, 2006, 2007). For the 2005 and 2006 edition of the conference, extensive submission and evaluation data exists. For the 2007 edition some of the evaluation data from the conference, particularly that of reverse engineered SCU files, were not completed at the time of these experiments. This unfortunately rendered the 2007 data considerably less useful. Given the presence of two years worth of useful data, the 2005 data will be used for training, tuning and optimizing the new summaries produced by my system while the 2006 will be used exclusively for evaluation.

In order to tune a system, a measure is required to tune the system to. For this work, the reverse engineered SCU data file was used to count the total SCU weight within a summary. This measure was chosen from three common measures: manually evaluated responsiveness, ROUGE scores, and SCU scores.

Manually-evaluating responsiveness was ruled out because manual methods are very labour intensive, and can be inconsistent. All manual evaluation for this work was performed by

volunteers so it was therefore not possible to have them examine dozens of system permutations in any level of detail which would produce a worthwhile analysis.

ROUGE scores were not chosen for three reasons. The first reason was that ROUGE is based on the idea of its scores correlating well with manual evaluation scores. This however does not imply that a high ROUGE score results in a high manual score. The second reason was that since ROUGE is a fully automated statistical measure, it is harder to determine if slight improvements in summaries actually correspond to a summary a reader would like better or simply an improvement in ROUGE score. A third reason not to choose ROUGE was that several different ROUGE measures exist, along with numerous combinations parameters for each measure. Throughout the past few years of DUC, ROUGE has itself been as much under evaluation as the systems it is evaluating. To this date, several ROUGE evaluations are conducted at DUC. There does not appear to be any consensus that a particular ROUGE measure is considered best.

The third method, SCU scores, was chosen because it provides an excellent, human generated measure, which can be made automatic by reverse engineering the past summary submissions. As a result, this measure can be easily computed for the summaries which were produced using a test system configuration without any significant manual labour.

Using the SCU scores does have a disadvantage regarding data availability. Since the evaluation at DUC was conducted by a limited number of volunteers, SCU data is not available for all topics and all systems. Only the systems from research groups and companies which volunteered to assist with the evaluation were evaluated using summary content units. Furthermore, given the limited amount of labour available, only a subset of the topics was evaluated. Even with these limitations, we still considered SCU scores the best option for evaluation.

There was one data-specific limitation in the SCU data. Since the SCU files were generated though reverse engineering of the evaluation files, only sentences appearing in summaries submitted to DUC will have SUC scores. This is not likely to be a major limitation on the positive set (sentences fitting well in the summary) given that most sentences of high utility to a summary will likely be picked by at least one system. In terms of the negative set (sentences which do not fit well in a summary), there is a fair bit of uncertainty since the

majority of sentences do not appear in any summary and in the majority of the cases with good reason. For example, there are many sentences which do not match any key word or synonym of a key word from the topic. Most of these sentences have little or nothing to do with the topic. On the other hand, the negative set could include summary worthy sentences which simply were not selected by any system at DUC. This is not a major limitation when tweaking a summarization system manually; however it could cause considerable problems with some machine learning algorithms. The negative set is unfortunately very inaccurate and could certainly confuse a classifier.

The summarization system described later in this thesis was tuned to SCU scores using two classes of methods: manual tuning and machine learning algorithms combined with some manual tuning. The manual tuning involves combining features and properties in using various mathematical means to achieve superior SCU scores. This tuning was performed using the UDC 2005 data.

In terms of machine learning, a variety of classifiers were used in an attempt to filter out sentences which are unlikely to contain at least one SCU. The goal is to be left with a set of sentences to select from where a greater percentage of sentences contain one or more SCUs. The result of this would be an increase in the probability that any summarization method would select sentences containing at least one SCU, than they were with the raw collection of sentences.

For the final evaluation, the 2006 DUC data was used. Summaries of at most 250 words were generated using the 2006 topics and the system tuned using the 2005 data. The 2006 DUC data contained 50 topics. Within those 50 topics, 20 topics had SCU data available. Consequently the core evaluation was conducted using those 20 topics. Three types of evaluation were conducted.

The first was mean modified SCU score. These results were produced using the reverse engineered SCU corpus, and were computed using the same formulas as the DUC evaluation. Therefore, the totals are comparable to the DUC SCU evaluation.

The second type of evaluation was conducted using ROUGE. This evaluation is also comparable with the evaluation conducted in DUC (see Figure 3.1) (DUC 2005). Since the system has not been in anyway tuned for ROUGE, it is not expected that the system will perform very well on this evaluation. A number of systems submitted to DUC may have been tuned to

this measure; therefore the methods we tested may not perform as highly in a ranking of ROUGE scores.

ROUGE-1.5.5.pl -n 2 -x -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -d

-n 2	compute ROUGE-1 and ROUGE-2
-x	do not calculate ROUGE-L
-m	apply Porter stemmer on both models and peers
-2 4	compute Skip Bigram (ROUGE-S) with a maximum skip distance of 4
-u	include unigram in Skip Bigram (ROUGE-S)
-c 95	use 95% confidence interval
-r 1000	bootstrap resample 1000 times to estimate the 95% confidence interval
-f A	scores are averaged over multiple models
-p 0.5	compute F-measure with alpha = 0.5
-t 0	use model unit as the counting unit
-d	print per-evaluation scores

Figure 3.1: ROUGE Parameters Used at DUC

The third type of evaluation is conducted manually. At DUC a manual evaluation was conducted by evaluators hired by the organizers, the National Institute of Science and Technology (NIST). To replicate this human evaluation as closely as possible, a similar local evaluation was conducted with a few minor changes.

The first change was that the local evaluators were volunteers. For this reason, the quantity of work needed to be limited. The DUC evaluation was conducted on all fifty topics. In our smaller-scale evaluation each evaluator was asked to evaluate six topics. To increase the total number of topics evaluated two teams of evaluators were used, thus doubling the number of topics evaluated to twelve. The twelve topics were chosen at random from the twenty topics for which SCU data existed.

Since the local evaluators used in this work where not the same evaluators as NIST used, the local evaluators evaluated one additional set of summaries from each topic. The additional set of summaries chosen for this additional evaluation was produced by the top-scoring DUC submission in terms of pyramid score. This additional evaluation will then serve as a basis for comparison between the scores assigned by the evaluators of this work and those hired by NIST.

In order to ensure there is no bias created by the local evaluators, for each topic three summaries were presented to read and evaluate. The first summary was the top performing

configuration of this system, the second summary was the baseline lexical matching system, and the third was the top performing system from DUC in terms of pyramid score. (see section 4.2.1) It was not revealed to the evaluators which summaries were produced by which system. Additionally, the order which the summaries were presented to the evaluators was varied between topics (see Appendix B). Since the baseline system and the system using features from the Connexor software were similar, some evaluators did report being about to tell those systems apart from the former DUC submission system. They were, however, not able to distinguish which summary was the baseline and which used the additional features.

In total, eight local human evaluators were used. Within this group, five of the evaluators had previous experience evaluating automatic summaries. For the other three, the task was entirely new. Every evaluator had a minimum of a bachelor's level university degree. The evaluators were not paid.

The evaluators were divided into two groups of four. Each group evaluated a different set of topics with all members of each group evaluated the same topics. Within each topic, different members of each group received the summaries in a different order. This was done because there were three summaries on each topic and an evaluator may recall facts from one summary to the next and may become confused on what facts were presented in which summary. Furthermore to this, the order of the summaries, in terms of which system created them, was varied between topics.

The twelve topics evaluated were chosen at random and divided between the two groups of evaluators. The division was conducted in a way that ensured a variety of topics went to each group. For example there were two plane crash topics, so one went to each group. For a list of the chosen topics see Appendix C.

The local evaluators were given a similar set of instructions (see appendix A) as the NIST evaluators. There was one question deleted from NIST's evaluation and one question added. The NIST evaluators were asked to assign an overall responsiveness score for each system. This score was produced after reading and evaluating all the summaries that a given system produced. The NIST evaluators were not given access to the scores assigned to the individual summaries when they answered this question. This evaluation was not feasible for the local evaluation for several reasons. The first reason was that the local evaluations did not take place in an

environment where it was possible to control the evaluator's access to the scores they have already assigned. The second problem with this question is that it would have required the local evaluators to know which summaries were created by which systems. This information would have led to a bias in the evaluation, because they all knew the author of the work personally.

Evaluation of summarization is not always able to show systems as being clearly better than others. In addition to the plurality of evaluation measures and varied requirements and opinions on what makes a good summary, past DUC evaluations have shown that the differences between the evaluation scores of many systems are not statistically significant at any common statistical significance threshold. In an effort to create a distinguishing measure, the local evaluators were asked to simply rank the three summaries for each topic. They were asked to do this as though they were a user of these three automatic summarizers. They were forced to produce a ranked order and no ties were allowed. While this is arguably a slightly more ambiguous means of evaluation since no particular ranking criteria was specified, it does capture the way users of a summarization system or any other software system might perceive the system. A user's first opinion of a software system is likely to be a very broad one, since more particular details require more time and thought.

3.2 Generating Features from a Semantic Parser

A large part of the challenge of utilizing shallow semantic features to improve responsiveness in automatic summarization is determining what information provided by a parser might be useful and finding efficient methods of extracting that information from complex parse structures.

3.2.1 Machinese Semantics Parser/Analyzer

Connexor Machinese Semantics (Connexor, 2003) is a semantic analyzer that provides semantic role recognition as well as grammatical, lexical, and sentential semantic features. The tool's output is similar to that of (some) text parsers in that the overall output is in the form of a tree (or forest). Attached to the nodes of the tree are recursive feature-value pairs containing semantic information. The full tree along with all the additional semantic information is encoded in XML (Extended Markup Language).

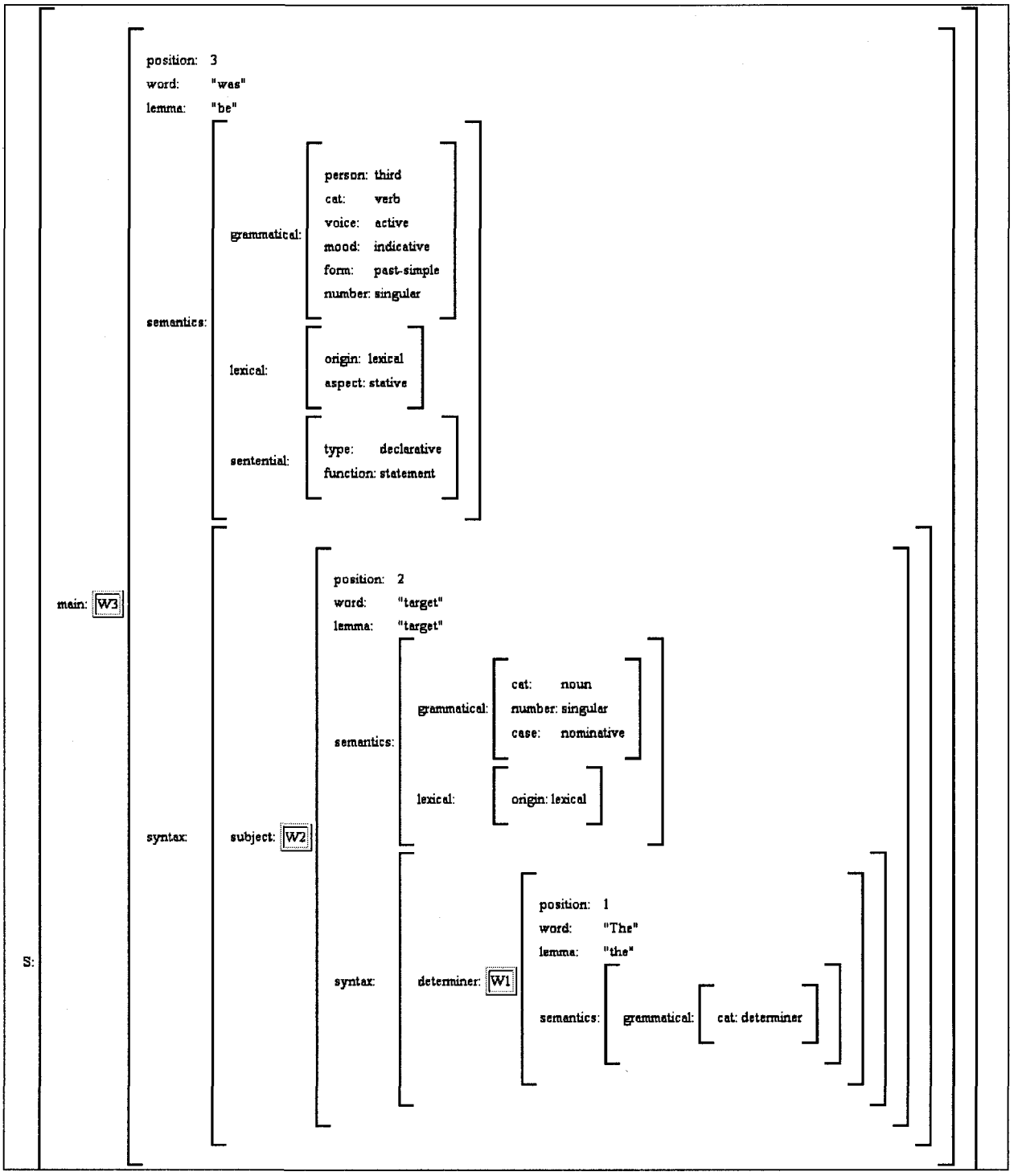

Figure 3.2: Example Parse (part I)

The target was, sadly, well-chosen.

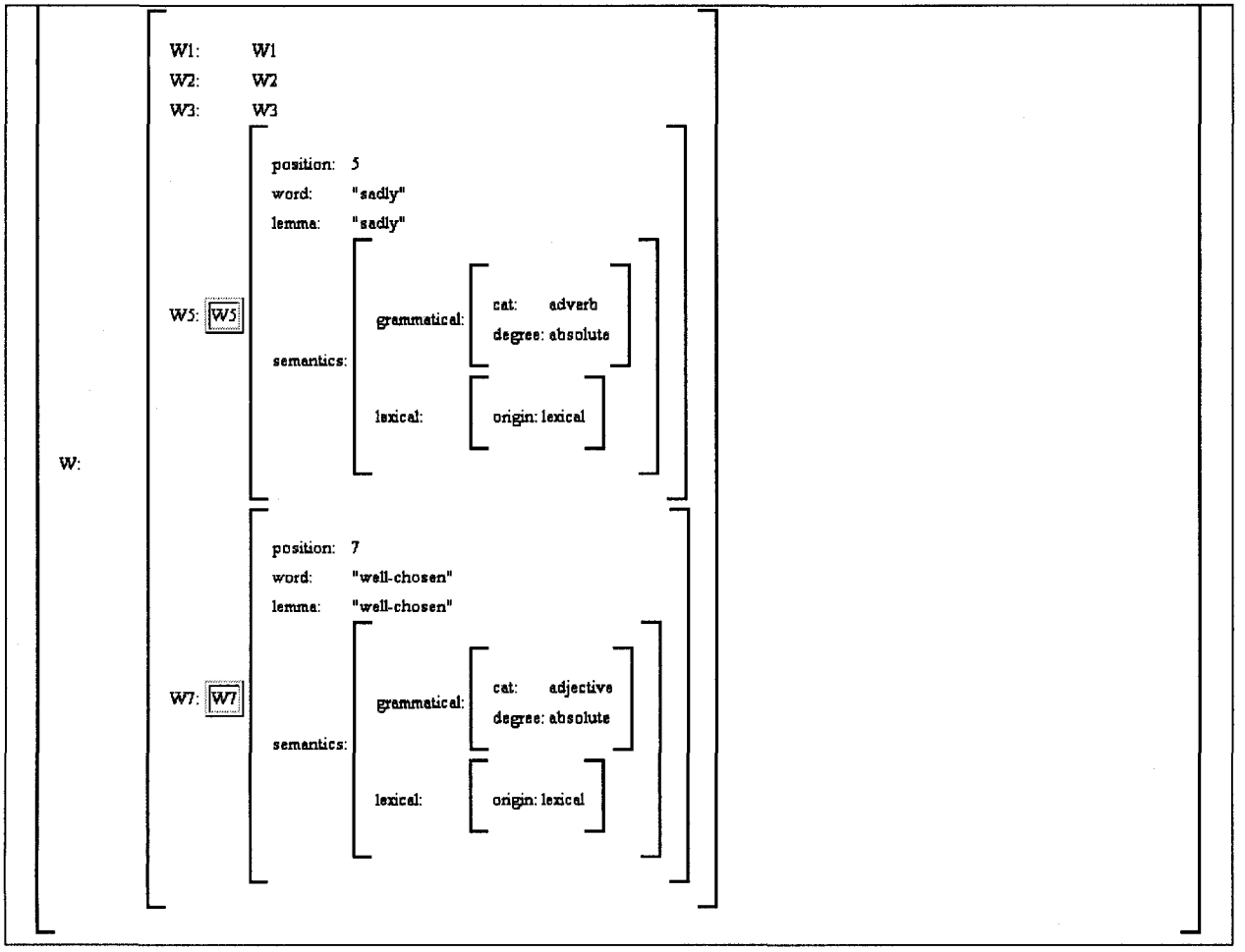

Figure 3.3: Example Parse (part II)

The target was, sadly, well-chosen.

3.2.2 Extracting Features

Machinese Semantics produces a deeply nested feature-value structure. Given that there is essentially no limit on how deeply the feature value pairs can be nested, it proved to be a non-trivial task to extract particular features or patterns of features from the structures. The task was further complicated because there was no common pattern to the overall structure. Particular features would only appear in situations where the analyzer was able to identify them. For some sentences a single parse tree was produced. In other cases, only a portion of the parse appeared in the main tree and some other clauses and single lose words were left separate.

The first issue in extracting features from such a diverse collection of parsers was to be able to study example parses. The deeply nested nature of the XML output makes

reading and comprehending the structure difficult. An XSL (Extensible Stylesheet Language) style sheet was created (Appendix D) to produce a more readable form. The style sheet is designed to display the parses in a standard feature-value bracket form commonly seem throughout NLP and linguistics. See Figure 3.2 and Figure 3.3 for an example of the style sheet output.

The second issue in extracting useful features is accessing them programmatically. The nested structures are very generic and can therefore be processed using a recursive algorithm. The complexity comes from dealing with the variety feature combinations encountered throughout the process.

In general, the parses did have the general structure of a collection of trees. Each node within each tree is a word. Each word structure contains the word itself, its position in the sentence, a lemmatized form of the word, and a head word for multi-word units. Along with the above listed basic information on each word, a series of semantic, syntactic, and linear syntactic elements are attached. The linear elements form things such as verb-chains. The semantic elements fall into three categories: lexical, grammatical and sentential. They are described below. In the case of the syntax elements and the linear syntax elements, they take the form of a sub-tree of word structures connected by a relation.

Each parse was stored as a nested class. This nested class structure was built using the Java programming language. The resulting class structure and parsing processing routines are entirely portable to other tasks outside of text summarization. The class structure maintained the same tree structure as the underlying parses. However, unlike a conventional tree structure, the class structure allowed particular properties of the parse to be passed up to the root of the structure for faster processing.

The main building block for storing the parses is a class for storing words and all semantic information attached. For each piece of semantic information a customized class is used for storing information. This allows information to be queried without time-consuming string comparisons at the time of processing for summary production. To this end a total of 21 classes were defined to hold this information.

The application programming interface (API) contained a set of standard functions including constructors and functions to ensure the data input is correct within each class. The classes also contained functions relevant to each attribute

3.2.3 Java Classes for Machinese Semantic Features

To store and process features, a total of 21 Java classes were defined. These classes performed a few basic functions such as verifying the data read against the set of possible values for each attribute. These classes' structures also stored all information relevant to each attribute and provided a variety of functions which were useful to process and retrieve information. These helper functions differed in each class depending on what information was applicable to that attribute. We begin with a brief description of each available attribute and some of its particularities.

Animateness

The animateness attribute is a simple true or false attribute. The field is true for the humans and animals and false for plants. It is possible to expand this using Connexor's custom lexicon feature. This was not explored in the scope of this work.

In terms of building an API, the animateness classes contained a simple getter function returning a Boolean value.

Grammatical Case

There are three possible values for the grammatical case attribute: nominative, accusative and genitive. They are described later. The grammatical case provided a Boolean function for each possible case type.

Part of Speech Category

This class stores the part of speech attribute for each word in the sentence. The class provides two methods of data access. The first is a simple function which returns the category as a string. The second method is a series of Boolean functions allowing a program to query a particular part of speech category. For example: is the word a noun?

Grammatical Degree

There are three possible values for the grammatical degree attribute: absolute, comparative, and superlative. This class provides Boolean get functions for each possible class type. A get function that returns the grammatical degree as a string is also made available.

Verb Form

The verb form was the most complex storage class. There are 27 possible types of verb forms. Each of the 27 possible types has two hypercategories: tense and aspect. Machinese Semantics maps each of these hypercategories to a pair of binary attributes. For tense, the binary features are "past" and "future". For aspect, the binary features are "perfect" and "progressive". For the infinitive categories there are only aspectual features. The verb form class stores the verb form information in 4 integer variables. These variables can take one of three possible values: 1 for true, 0 for false, and -1 for situations where the feature is not present.

This design allows this feature to be queried in more than one way. The most obvious is to query for the verb form itself and have the result returned in a string. The other method is to query the temporal features directly. This greatly increases the flexibility for use and fast processing.

Grammatical Gender

The grammatical gender attribute is relevant to certain pronouns and determiners. It has two possible values: "masculine" and "feminine". This class is designed to allow each of these to be queried separately with a binary result.

Lexical Semantic Class

There are seven possible values for the lexical semantic class attribute: "human", "female", "male", "nationality", "event", "animal" and "plant". The storage class is designed to allow direct binary queries for each class type as well as a string query for the name of the type. This feature is highly dependent on using a custom lexicon.

Location

The semantic parser can tag five possible types of locations: countries, cities, regions, website addresses, and general locations. The ability to query for a location as a string as well as a binary get function for each type was provided in the design.

Grammatical Mood

A binary get function was provided for each of the four possible grammatical moods identified by the Connexor semantic parser. An additional get function could also provide the name of the mood as string. The possible types were: "indicative", "subjunctive", "conditional", and "imperative".

Grammatical (Morphological) Number

This attribute as two possible values: "singular" and "plural". This value can easily be stored in a binary fashion. A pair of Boolean functions is provided to access this feature. The first function returns true if the attribute is set to singular. The second function returns true if the attribute is set to plural. Although both functions clearly return the logically negated result of each other, both were provided to slightly increase readability of program source code.

Organization

The organization storage class utilizes a similar design as was used for location and the lexical semantic class. An access function which returns a Boolean value is provided for each of the five possible categories of organizations. They are general organizations, associations, companies, stock exchanges and governments.

Origin

The information stored in this class is not the usual semantic, syntactic, or lexical information provided by the parser. This attribute keeps track of where words were found by the parser. This includes the parser's main lexicon, its heuristic component or its custom lexicon. A Boolean get function was provided for each of these types. Since the custom lexicon was not part of this work, this feature is of little use but it was included for future uses.

Grammatical Person

The parser identifies the three categories for person in verbs pronouns and determiners within the English language. The possible values are first, second and third person. The class provides a Boolean function to test whether a particular word falls into each category.

Proform Type

There are eight proform types identified by the parser. These are relevant for pronouns and certain determiners and adverbs. A binary function was provided to query each type as well as a function returning a string to access the name of the type.

Proper Noun

This is a simple attribute indicating if a noun is a proper noun. A simple Boolean function indicates so. There is no obvious way this attribute assists summarization directly but it could easily assist an auxiliary task such pronoun resolution in either post processing or pre-processing.

Refining

The refining attribute is a lexical semantic attribute indicating whether something is a material, tool, cloth, product or technology. A Boolean function was provided to determine if a word was one of each. To make effective use of this attribute a custom lexicon would be required.

Sentence Function

There are eight sentence functions identified for the main verb of a sentence or clause. They are described in detail below. In addition to a Boolean function for each sentence function, an additional Boolean function is provided to determine whether or not the sentence functions as a question.

Sentence Modality

The sentence modality feature identifies if the main predicate of a sentence or clause denotes an obligation or describes an action that is possible. A Boolean get function was provided for each situation.

Sentence Type

The sentence type feature identifies four sentence types: interrogative, declarative, imperative and explanative. An access function which returns a binary value is provided for each. Additionally, a function which returns the type as a string value is included.

State

The state attribute identifies if something is a solid, liquid or gas. A storage class for this attribute was included for this attribute in case these classes are being used in another application. It is unlikely there is a summarization use for this attribute.

Grammatical Voice

There are two voice types that the Connexor Machinese Semantics parser identifies. The first is an active voice the other is a passive voice. This attribute comes attached to verbs. A Boolean get function is provided for each. A function returning a string value is also included within this class.

3.2.4 Accessing Attributes

The main consideration in accessing attributes is the ability to process the parse trees efficiently without consuming too much runtime. Since all the sentence structures are stored in the form of a tree, it can become a time-consuming task to traverse all the branches looking for particular attributes. This was dealt with several ways in the design.

The first method of improving access time was to produce word lists. This allows a linear time traversal of the words of a sentence, in the order of occurrence within the sentence. The lists could also be limited to only certain words from the sentence. The parse produced by the semantic analyzer, is a collection of trees (or a forest). As a result, words which were adjacent in the original sentence do not always appear within the same trees. This makes a task like matching bigrams considerably more difficult since there is a great deal of looking around for adjacent works within different trees. While bigrams could be produced using the original sentence, the parsed version has certain possible advantages such as lemmatized forms of the words. The word lists are produced at the time of the parse or when reading the parse from a storage file.

Word lists are also used to gather certain types of words used in the creation of some features. Examples of these are nouns, verbs, locations and nationalities. The major speed increase is due to the fact that a list, once produced, can be reused several different times without having to traverse the tree again a second time to produce it.

It may not initially appear as though creating lists of words in the sentence would be faster to access than they would be by performing a tree traversal. The complexity of Machinese Semantics means that the trees do not only contain words as nodes but also other relationships between words and portions of the structure. This consequently takes longer to traverse than a tree consisting of only words as nodes.

The second method of improving access time is to define functions at the sentence level that link directly to functions located within the tree. An example of this would be a function to determine if the main verb of a sentence is in first person. Such a function would have a direct link, using a variable populated when the parse is processed, to the main verb of the sentence. At this point it simply has to call the method to determine if the verb is in first person. In the event that the parser was unable to identify a main verb for a given sentence, this sentence level function would be able to identify this quickly because the access variable is not populated. The performance enhancement of the second method is that a tree traverse is no longer required in order to find particular pieces of information. All of the required lists and direct access variables are able to be populated during a single tree-traversal at the time the tree structure is read. This is an improvement over performing a separate traversal each time information is required.

3.2.5 Choosing Features for Summarization

With the wide variety of features made available by the semantic analyzer, the next requirement is to determine which features would be useful to automatic summarization. Additionally, we must determine exactly how these features could be used. In order to achieve this, we first consider what features might help chose which sentences to select for inclusion in a summary. Next, we conduct a frequency analysis using 2005 DUC data. The purpose of this analysis is to determine if a feature is present disproportionately in summaries created manually compared with automatic summaries and the source document set. In particular we are looking for certain types of attributes appearing more frequently in

model summaries than in machine generated summaries. The Machinese Semantics semantic feature-value pairs fall into three broad categories: grammatical, sentential, and lexical.

To select which features to use from the semantic analyzer, all the features were considered individually with the following question: *If this was the only feature available to select sentences for a summary, would it be at least minimally helpful?* It must be remembered that a computer has no actual understanding of sentence content. It ultimately makes choices based on certain numbers derived from sentences. It must further be noted that, like in many problems in natural language processing, it is very easy to find exceptions to generated rules. In order to make progress, we must either be able to find ways to identify these exceptional situations or be able to ignore them in favour of satisfying the most common scenarios.

We will use lexical matching as our base statistic for ranking sentences. We will conduct a brief experiment to choose the best variant. These variants will make use of some information made available by the semantic analyzer. All additional features will then be added to this base statistic. Features will be individually added to the base statistic to gain a measure of how they perform individually. They will then be grouped and weighted to gage how they interact together. As more features were added, the weights of some previously added features often required adjustment since features can interact with each other in various ways.

3.2.5.1 Lexical Matching Using Only Certain Parts of Speech

The part of speech category classifies each word in the manner that a part of speech tagger does. An experiment was conducted into whether this information could be used to improve the lexical matching that most query based system use in some form or another.

Two common methods of performing lexical matching within natural language processing are straight word by word lexical matching or lexical matching with stop words removed. In the first case all words in the query are matched with each word in the candidate sentence. The score is based on how many words in the sentence match words in the query. In the later case the score is calculated similarly except that words appearing in Sanderson's (1994) list of stop words did not count towards the match score.

In terms of utilizing part of speech tagging, the score is calculated in a similar fashion as with stop word removal except that instead of removing stop words, words tagged as certain parts of speech are removed. For this test, the parts of speech included in the matching were: nouns, verbs, adverbs, numerals, pronouns, interjections, and adjectives. The DUC 2005 data was utilized for this test. In all cases the lemmatized form of all words was used. This information was also provided by the Connexor tool. The score was computed using two measures. The first measure, SCU count, was the total number of SCUs contained in the summaries for all topics (for which SCU data exists) in DUC 2005. The second measure, weighted SCU count, was weighted SCU total for all SCU topics. The SCU weights were determined by the number of model summaries each SCU appeared in. The results of this experiment appear in Table 3.1.

In both measures it was determined that limiting the lexical matching to certain parts of speech was helpful. It did not however achieve nearly the performance of lexical matching with stop words removed. Consequently, lexical matching with stop words removed was chosen as the lexical matching mechanism for the system. This form of lexical matching also forms our baseline which we will compare performance against when other features are added.

	Standard lexical match	**Lexical match with stop words removed**	**Lexical match on limited parts of speech**
SCU Count	100	152	108
Weighted SCU Count	262	387	267

Table 3.1: Comparison of Lexical Matching Limited by Stop Word Removal and Parts of Speech

3.2.5.2 Grammatical Semantic Features

The first category of Connexor features, Grammatical Semantics, contains up to 10 features which can appear where applicable. They are listed in Figure 3.4. Five features from this category where examined as possibly having relevance to summarization.

Part of Speech Category	Verb form
Morphological Number	Grammatical case
Grammatical person	Grammatical gender
Grammatical voice	Grammatical mood
Grammatical degree	Proform type

Figure 3.4: Machinese Semantics Grammatical Semantics

Verb Form

As mentioned before, there is a feature in the Connexor Machinese Semantics for the tense of all words tagged as a verb. This feature classifies verbs as belonging to one of 25 categories (see Figure 3.5). Additionally, each of these categories is explained using four hypercategories: past, future, perfect, and progressive.

The hypercategories were used to develop features useful for summarization. The rational for looking at the hypercategories is that the analysis becomes considerably simpler. With a limited amount of data available, it was considerably more feasible to deal with only four categories rather than using twenty-five. In the later case, the data would become very sparse to cover all categories well. In order for the analysis not to become too complex, only words marked by Connexor as main verbs were examined. A major reason for this simplification is the amount of test data available from DUC is fairly limited. This limited quantity of data makes it very hard to test rules which are complex in nature because there are not enough examples within the training and test data. The second practical problem is that sentences are being used as the elementary unit for extraction. Attempting to use verbs within secondary clauses would greatly increase the complexity of the task, particularly in situations where the verb tense of a verb in a secondary clause conflicts the tense of a verb in the a main clause.

The first thing to examine in the development of verb-related features is the frequencies of the various verb tenses appearing throughout the 2005 DUC data. The sentences in the source document have a certain frequency of occurrence of these verbs tenses. If we compare these frequencies with the frequencies of the verb tenses in the sentences selected by the community (the peer summaries), we find that no particular verb tenses were favored in the selections made by the summarization systems. The largest

differences were in the perfect verb tense, where the community slightly favored sentences with that hypercategory set to true. See Table 3.2Table 3.5 for details.

Category	**Explanation**
present-simple	past: F future: F perfect: F progressive: F
present-perfect	past: F future: F perfect: T progressive: F
present-progressive	past: F future: F perfect: F progressive: T
present-perfect-progressive	past: F future: F perfect: T progressive: T
past-simple	past: T future: F perfect: F progressive: F
past-perfect	past: T future: F perfect: T progressive: F
past-progressive	past: T future: F perfect: F progressive: T
past-perfect-progressive	past: T future: F perfect: T progressive: T
future-simple	past: F future: T perfect: F progressive: F
future-perfect	past: F future: T perfect: T progressive: F
future-progressive	past: F future: T perfect: F progressive: T
future-perfect-progressive	past: F future: T perfect: T progressive: T
future-anterior	past: T future: T perfect: F progressive: F
future-perfect-anterior	past: T future: T perfect: T progressive: F
future-progressive-anterior	past: T future: T perfect: F progressive: T
future-perfect-progressive-anterior	past: T future: T perfect: T progressive : T
simple	(subjunctive) simple
progressive	(subjunctive) progressive
simple-infinitive	perfect: F progressive: F
perfect-infinitive	perfect: T progressive: F
progressive-infinitive	perfect: F progressive: T
perfect-progressive-infinitive	perfect: T progressive: T
perfect-participle	perfect: T progressive: F
progressive-participle	perfect: F progressive: T
perfect-progressive-participle	perfect: T progressive: T

Figure 3.5: Machinese Semantics Verb Forms

The first thing to examine in the development of verb related features is the frequencies of the various verb tenses appearing throughout the 2005 DUC data. The sentences in the source document have a certain frequency of occurrence of these verbs tenses. If we compare these frequencies with the frequencies of the verb tenses in the sentences selected by the community (the peer summaries), we find that no particular verb tenses were favored in the selections made by the summarization systems. The largest differences were in the perfect verb tense, where the community slightly favored sentences with that hypercategory set to true. See Table 3.2Table 3.5 for details.

Past	**Document Set**		**Peer Summaries**		**Model Summaries**	
	Count	**Percent**	**Count**	**Percent**	**Count**	**Percent**
True	16944	41.6%	4954	41.4%	1644	44.0%
False	23818	58.4%	7016	58.6%	2093	56.0%
Total tagged	40762		11970		3737	

Table 3.2: Frequency of Past Tense Main Verbs in Peer and Model Summaries vs. the Source Document Set

Future	**Document Set**		**Peer Summaries**		**Model Summaries**	
	Count	**Percent**	**Count**	**Percent**	**Count**	**Percent**
True	2175	5.3%	672	5.6%	75	2.0%
False	38587	94.7%	11298	94.3%	3662	98.0%
Total tagged	40762		11970		3737	

Table 3.3: Frequency of Future Tense Main Verbs in Peer and Model Summaries vs. the Source Document Set

Perfect	**Document Set**		**Peer Summaries**		**Model Summaries**	
	Count	**Percent**	**Count**	**Percent**	**Count**	**Percent**
True	3972	9.7%	1403	11.7%	359	9.6%
False	36790	90.2%	10567	88.3%	3378	90.4%
Total tagged	40762		11970		3737	

Table 3.4: Frequency of Perfect Tense Main Verbs in Peer and Model Summaries vs. the Source Document Set

Progressive	Document Set		Peer Summaries		Model Summaries	
	Count	Percent	Count	Percent	Count	Percent
True	1707	4.2%	635	5.3%	180	4.8%
False	39055	95.8%	11335	94.7%	3557	95.2%
Total tagged	40762		11970		3737	

Table 3.5: Frequency of Progressive Tense Main Verbs in Peer and Model Summaries vs. the Source Document Set

Looking at the frequencies for the model summaries where the sentences were hand-written rather than extracted from the document set; there are a noticeably lower proportion of sentences in the future tense.

There are a few explanations as to why this may be desirable. The first is found in the topic statements. Most topic statements ask for a report on something that has already occurred or for information on the developments in some situation. All of these cases favor information from the past. The second problem with the use of the future tense is that all the information in the summaries, either extracted in the case of the peers or re-written in the case the of the models, has been taken from news articles which have been written, at a minimum, a number of years ago. A verb in a future tense, written in the distant past may no longer make sense when read in the present. At the very least it may introduce ambiguity or confusion.

If a sentence such as *George Bush will run for president in the next presidential election*, written in the year 2000 is read in the year 2008 it would carry a different meaning since we have confused what election is being referred to.

The final verb tense features contain the binary values for each of the four hypercategories. In the development of the final heuristics to produce the summaries, these findings can be taken into account in terms of how certain tenses are weighted.

Grammatical Case

The parser identifies three grammatical cases for nouns and pronouns. They are nominative, accusative and genitive. These terms are briefly described by Loos et al. (2004). A noun is often tagged as nominative when it is the subject of a sentence. The accusative nouns usually appear when a noun is the direct object. The genitive case generally appears when the noun or pronoun is showing possession.

An examination of the frequencies of occurrence with the 2005 DUC data (see Table 3.6) reveals the parser did not mark any words as accusative. Within the nominative and genitive classes, the peer summaries selected words at essentially the same frequencies of occurrence as in the document set. The model summaries did slightly favour nominative forms rather then genitive.

Although the difference is small, and may not make a major difference in summaries, there is enough of a difference to warrant the creation of features for summarization. A resulting heuristic could slightly favour nominative forms. There are a few possible reasons why summaries might present these numbers. The lack of accusative could be explained by either an incorrectly functioning parser feature or simply a lack of them in the text. In terms of the change in frequency between the nominatives and the genitives, it does make some intuitive sense since most summarization topics talk about situations and events. Both of these types of discussions are perhaps less likely to produce situations where possessives would be required. This could certainly vary greatly depending on topic.

Grammatical Case	Document Set		Peer Summaries		Model Summaries	
	Word Count	Percent	Word Count	Percent	Word Count	Percent
Nominative	306002	97.27%	118668	97.38%	26546	98.11%
Accusative	0	0.00%	0	0.00%	0	0.00%
Genitive	8593	2.73%	3194	2.62%	511	1.89%

Table 3.6: Frequency of Grammatical Case Nouns and Pronouns in Peer and Model Summaries vs. the Source Document Set

Person

For the person attribute, we look only at the main verb of a candidate sentences. This avoids possible conflicts with verbs in sub-clauses. There are some noticeable differences in frequency of occurrence within the 2005 DUC document set.

In all cases, verbs in the 3rd person form are most prominent (see Table 3.7). Given that the data source was news articles, this makes intuitive sense. In most news articles, first and second person are only used in direct quotes or opinion articles. These types of sentences could be bad for a summary. Sentences in the first person would be a disaster if it

is not clear who the speaker is. Likewise, a sentence written in the second person could cause considerable problems since it is not known who the addressee is. In a summary, it is unlikely that there sufficient space for direct quotes or at least not many.

Person	Document Set		Peer Summaries		Model Summaries	
	Count	Percent	Count	Percent	Count	Percent
First	1432	4.4%	132	1.3%	5	0.2%
Second	290	0.9%	88	0.9%	16	0.5%
Third	30599	94.7%	9861	97.8%	3095	99.3%
Total tagged	32321		10081		3116	

Table 3.7: Frequency of Person in Main Verbs in Peer and Model Summaries vs. the Source Document Set

It is notable that within the model summaries very few first and second person sentences are used. This feature will consequently give a slight boost to sentences where the main verb is in the 3rd person, and slightly penalize those where it is not.

Grammatical Degree

The semantic parser marks grammatical degree for adjectives, adverbs, determiners and pronouns. There are three possible classes identified: absolute, comparative and superlative.

Across the DUC 2005 document set, most words were tagged with this attribute were classified as having the absolute degree (see Table 3.8). The frequencies of words tagged to each class within the peer and model summaries fall along fairly similar lines as frequencies of word tagged to each class in the document set. The model summaries had a slightly higher proportion of absolute degree words and a somewhat lower proportion of superlative degree words.

Grammatical Degree	Document Set		Peer Summaries		Model Summaries	
	Word Count	Percent	Word Count	Percent	Word Count	Percent
Absolute	72281	95.54%	26303	95.29%	6191	96.25%
Comparative	1892	2.50%	609	2.21%	154	2.39%
Superlative	1479	1.96%	692	2.51%	87	1.35%

Table 3.8: Frequency of Grammatical Degrees in Adjectives, Adverbs, Determiners and Pronouns in Peer and Model Summaries vs. the Source Document Set

This makes some intuitive sense because a superlative specifies some sort of extreme or outlier. This may not be desirable in a summary since in the limited space of a summary it is usually advantageous to talk about a regular case rather than extreme cases which could simply be exceptions.

Grammatical Mood

Four grammatical moods are identified by the semantic parser: indicative, subjunctive, conditional, and imperative.

The 2005 peers selected sentences with approximately the same frequency of moods as occurred in the source document set. The model summaries marginally favoured sentences with indicative verbs forms, although most sentences are in this form. There was a considerable drop in the frequencies of all other moods.

Grammatical Mood	Document Set		Peer Summaries		Model Summaries	
	Word Count	Percent	Word Count	Percent	Word Count	Percent
Conditional	809	1.98%	260	2.16%	38	1.01%
Imperative	339	0.83%	97	0.81%	18	0.48%
Indicative	39446	96.61%	11600	96.55%	3685	98.32%
Subjunctive	238	0.58%	57	0.47%	7	0.19%

Table 3.9: Frequency of Grammatical Mood Classifications in Peer and Model Summaries vs. the Source Document Set

In terms of how these terms might apply to a summary, it is very hard to imagine many situations where imperatives, which provide commands or orders, would be useful in a summary. The topics that appear within the DUC data sets look for information or facts. It is harder to hypothesize why conditional and subjunctive moods do not appear more in summaries. One possible explanation is simply that words of the indicative mood are better for providing the type of facts the summary topics are looking for.

3.2.5.3 Sentential Semantic Features

The second category, sentential semantics, contains three features. They are listed in Figure 3.6. From this set of features, two were deemed to have a possible relevance to automatic summarization. This determination was made intuitively based on what type of features might help extractive summarization.

Sentence modality	Sentence function
Sentence type	

Figure 3.6: Machinese Semantics Sentential Semantic Properties

Sentence Type

Machinese Semantics uses four classes for sentence type: interrogative, declarative, imperative, and exclamative. Across the set of 2005 documents the parser did not identify any sentences as exclamative or imperative. Connexor's documentation (Connexor 2003) also did not provide any examples of these two types. The sentences in the document set and consequently the summaries were mostly declarative sentences with very few interrogative sentences appearing in the human written summaries. Interrogative sentences are all in the form of a question; therefore it is not surprising that few appear in the summaries. The likely reason for this is that the role of the summaries is to provide information or answer some form of direct or indirect question. In many cases, more questions do not provide this type of information.

Sentence Type	Document Set		Peer Summaries		Model Summaries	
	Count	Percent	Count	Percent	Count	Percent
Declarative	42596	99.24%	12780	99.34%	3793	99.71%
Exclamative	0	0.00%	0	0.00%	0	0.00%
Imperative	0	0.00%	0	0.00%	0	0.00%
Interrogative	328	0.76%	85	0.66%	11	0.29%

Table 3.10: Frequency of Sentence Types within Peer and Model Summaries vs. the Source Document Set

Sentence Function

There are eight classes for sentence type defined by the parser-analyzer. They are statements, commands/directives, exclamations, reporting clauses, tag questions, tone questions, wh-questions, and option questions.

In the frequencies of occurrence in the 2005 data, the question functions do not appear very often in the model summaries relative to the number of times they appear in the document set as well as peer summaries (see Table 3.11). This is intuitive since it unlikely that sentences asking questions will provide much information for a summary. Most identified sentence functions were statement. There was also a marginally larger proportion

of these used in both the model and peer summaries compared to the document set. Similarly commands are also less likely to work well in a summary. Command sentences are more useful in giving orders and are harder to conceive as a means of providing information. With this information statement sentences will be favored in subsequent heuristic development.

Sentence Function	Document Set		Peer Summaries		Model Summaries	
	Count	Percent	Count	Percent	Count	Percent
Tag Question	0	0.00%	0	0.00%	0	0.00%
Tone Question	11	0.03%	2	0.02%	0	0.00%
Wh Question	284	0.77%	61	0.54%	14	0.38%
Op Question	71	0.19%	16	0.14%	3	0.08%
Statement	36324	98.65%	11088	99.02%	3671	99.32%
Command	132	0.36%	31	0.28%	8	0.22%
Exclamation	0	0.00%	0	0.00%	0	0.00%
Reporting Clause	0	0.00%	0	0.00%	0	0.00%

Table 3.11: Frequency of Sentence Functions within Peer and Model Summaries vs. the Source Document Set

3.2.5.4 Lexical Semantics

The third category, lexical semantics, contains up to ten properties. From these we examine two: location and nationality. The latter is found within the semantic class attribute.

Origin	Aspect
Proper	Animate
Class	Refining
State	Organization
Location	Semantic Class

Figure 3.7: Machinese Semantics Lexical Semantic Properties

Location Correction

A common problem with lexical matching is that words that refer to either the same things or parts of the same things do not match directly. It is possible, in these situations, to

utilize lexical resource such as Wordnet (Miller, 1995) or Roget's Thesaurus (Jarmasz, 2003) to allow such words to match indirectly. The difficulty with such a process is that applied broadly across a large set of words can essentially create too much matching. For instance in summarization, only a very limited number of sentences can be present in the finally summary. Adding all synonyms and hyponyms to the matching process used to select sentences will simply return a huge set of sentences that can not appear in the summary. Consequently, it is necessary to be very selective about where to expand lexical matching.

The Connexor semantic analyzer is able to identify location names within sentences. Locations are example of words that exhibit a hierarchical property, where places are subsets of larger places. For example the city of Paris is inside France. The result of this is that a topic asking for information or examples within France will not be able to achieve a lexical match with a sentence containing only the word Paris.

The semantic analyzer can however identify both Paris and France as locations. We are then able to utilize the Wordnet hierarchy to match these locations together. For the same reasons as stated above, it is still necessary to utilize the expanded matching with care. In addition to identifying locations, the Connexor tool will also categorize them as a country, city, region, general location, or website.

With this information it is possible to limit the number of levels in which that we climb in the Wordnet hierarchy. For example when comparing a country name to a city name, we only need to allow for two hops. The first hop goes from a city name to a state or province name, and the second hop goes from a state or province name to a country name.

In general we need to allow a bit more than the minimum matching since the hierarchy and structures of countries and regions of the world can differ. The number of levels required to traverse the tree to find matches was determined using Wordnet's browser and examples from a variety of locations in the world. When comparing a country to a country we allow for two levels. In all other comparisons an allowance of four levels is used. The reason regions cannot be given special treatment is that they can refer to large regions or continents, as well as regions within countries. Locations marked as websites are not compared.

The topic terms are always considered as the superset and the words in the documents are considered the subset. The reason is that a topic dealing with North America can use examples from any place inside of North America. It is not impossible to conceive a topic statement where this logic would fail, however for such a topic this process would perform no worse than straight lexical matching, since no matching would be found in both cases.

Nationality Correction

The nationality correction is very similar to the location correction. An example of a problem that it solves is one where a topic asks for examples of something from within a country, using the nationality rather than the country name. For example: "*Discuss examples of Canadian hydro-electric projects.*"

In such a case the word *Canada* does not produce a match. As well, subset places such as province names or city names also do not help. Here it is necessary to first convert the nationality to its respective country name and then compare the result with subset place names. A difference between this process and the one used to compare locations is that nationalities in both the topic statements and the content sentences must be converted. The conversion between a nationality and a country name are found within Wordnet using holonym relations (member-of relations).

This process, like the one for location comparison has the advantage of producing no side-effects in situations where no nationalities are present or the system has failed to match a nationality.

3.2.5.5 General Features

Main clause

The Connexor semantic parser produces a parse tree which is heavily tagged with information. At the root of the parse tree for a complete parse is the main clause starting with the main verb. In cases of an incomplete parse, a forest is created rather than a single parse tree. There are two sub-cases for incomplete parses. Some incomplete parses will have a main parse tree containing a main clause; however, there are one or more pieces of

the sentence which could not be added to the main tree. The second type of incomplete parse produces a forest of trees with no main verb identified.

The first type of incomplete parse often occurs in situations where a sentence is long and complex and there are many nested clauses. There is often nothing wrong with these sentences. The parser simply had difficulty with them.

The second type of incomplete parse is more likely to occur in situations where the input is not a true sentence, or if it is a sentence with bad structure. These cases, particularly the first, are a result of things like titles and sub-headlines being captured from the source documents. Things like titles often produce lexical matches since they tend to contain many key words. However, it is not good to include them in a summary, since they are not complete sentences.

The main clause feature is designed to filter out sentences based on how well formed their parse tree is. The value of this binary feature depends on whether a main verb was found for a candidate sentence.

Features from Previous Work

In a previous work (Copeck et al. 2006), there are a number of useful features. These were an expanded set of features from those used in (Copeck et al. 2005). In order to be able to use training data from the 2005 edition of DUC, we were forced to stick with the more limited set of features used in 2005 submission. All of these features appeared in the set of features for 2006.

The features presented here are added to the statistical features from our existing system: number of characters in the sentence, number of words in the sentence, number of the paragraph in which the sentence appears, sentence sequence number in the paragraph, number of discourse connectives (Marcu, 2000) in the sentence, number of words in the sentence indicative of causality, number of proper nouns in the sentence, number of content words in the sentence, number of content bigrams, number of punctuation marks in the sentence, and total number of pronouns in the sentence.

Several of the features are similar to those we already have produced from the output from Connexor. We will add the basic features the number of words, number of characters,

paragraph number and sequence in paragraph to the feature set. We included these particular features since they are basic things which have often been shown to be helpful in summarization of newspaper articles. These features have been used in summarization systems dating back as far as (Edmundson, 1969).

We did not find that there was sufficient data on causality and Marcu connectives to perform sufficient experimentation alongside new features. Given the development of summarization systems such as (Copeck et al. 2006), which resolve pronouns, we left off proper nouns as a feature. Since we use word matching as our base statistic these features were redundant to what we already have and were therefore not included.

3.3 Discussion of Experiments

To add the new features to the system, two main methods are proposed. The first is to attempt to extend the existing heuristics to utilize some of the new features in a logical way. The second method uses all of the features with a machine learning algorithm to either perform sentence selection or assist in doing so.

3.3.1 Machine Learning

Machine learning was used to predict whether a given sentence contained a summary content unit (SCU). For these experiments the SCU-marked data from DUC 2005 was used as training data and the data from DUC 2006 was used for testing.

The first machine learning approach was to build a classifier using the 2005 data and then use it to select sentences for each summary in the 2006 data. Since the classifier may select too many sentences to fit in a 250-word summary, the classifier's output probabilities could be used to rank the selected sentences and ultimately determine what sentences are included in the final summary.

The second approach uses machine learning to pre-filter sentences. The rationale behind this second approach is that the output probabilities from a classifier represent how well a selection fits a particular class. This has little to do with how good a sentence is for a summary. For example a sentence just barely inside the threshold for a particular class

could in fact be excellent in a summary. The class selection only indicated that the sentence could be useful in the summary. For every topic there are many sentences that could go in a summary. By the nature of summarization, not all these sentences can be included in a particular summary due to space limits. Because of this, an additional process is required to determine which sentences make the final summary.

Such a process is likely to behave very similarly to a regular selection method not involving machine learning. In this case, the role of machine learning is to increase the chances that a selection algorithm chooses good sentences. This can be measured by whether the sentence contains any SCU.

Currently within the global set of sentences across all topics that have been tagged with summary content units, approximately 10% of sentences contain at least one SCU. Consequently, when selecting a sentence at random there is a fairly low chance of selecting one containing a SCU. The aim of this experiment is determine if a machine learning classifier can take, as an input, the complete set of sentences, approximately 10% of which contain a SCU, and output a subset of sentences, in which a larger percentage contains a SCU. This is defines a baseline for what precision score we would need to achieve.

With a resulting set of sentences, in which a greater percentage contain at least one SCU, any sentence selection algorithm has a greater chance of choosing sentences with at least one SCU.

3.3.2 Heuristics-Based Approaches

All heuristics are developed based on the idea that sentence selection is tied closely to lexical matching between words in the query sentences and words in candidate sentences. Consequently all heuristics developed are designed to either slightly enhance the count of matching words or penalize it.

In terms of the basic lexical matching, we used lemmatized forms of all words as provided by the parser. As determined in an experiment discussed earlier we required that the words not appear on a list of stop words.

The location and nationality features were designed to enhance a lexical match by matching words such as "Paris" and "France" that would not otherwise have matched. In

such a case there matches were simply added to the total of matching words the particular sentence.

For the sentence function feature the lexical match score for "statements" was enhanced by 2; "reporting clauses" where rewarded by 1; and all other types where penalized by 2. These factors where determined by conducting frequency analysis on submitted and model summaries from DUC 2005.

Similarly, for the sentence type feature, the lexical match score for declarative sentences was increased by 1, and all other types where decreased by 2. These factors where also developed through frequency analysis of past submissions and models.

Verb tense features are simply binary features what could be added, multiplied or subtracted from each heuristic. This structure provides the widest set of options for inclusion in a heuristic.

For grammatical person features, sentences written in the 3rd person received a score increase of 4. Sentences in the 1st person received an increase of 2 since they are mostly direct quotations. Sentences in 2nd person were penalized by 5. Sentences in 2nd person are very rare in newspaper articles and do not generally fit well in a summary.

The grammatical mood feature was used in heuristics based on reasoning as to what types of sentences would be most sensible in a news summary. Indicative sentences received a boost of 3. Subjunctive and conditional sentences received a boost of 1. The score for imperative sentences was decreased by 2 since they are not a good fit for a summary which is written to provide information to the user.

For grammatical degree, a statistical feature was developed based on the counts of words categorized by each type. Absolute words added 2 to this statistic since they are the clearest when standing alone. Comparative words added 1 to the statistic. Superlative words subtracted 1 from the statistics since they tend to leave more ambiguity when seen outside of a large context. The entire statistic was normalized based on the total number of all three words in the sentence.

The main clause feature is similar to the verb form feature since it is binary in nature. It can therefore be used to filter the sentences or to simply enhance or penalize the final score.

3.3.3 Combining Features

As noted in later sections, the features created from the semantic analyzer output do not combine well using automated mechanisms. A reason for this is that automatic mechanisms for combining features typically use linear or polynomial equations. Breaking anyway from automatic methods allows for far broader array of methods including filter certain sentences out based on certain features. The disadvantage to using manual tuning is that it becomes impossible to prove an optimal method for combining features. We feel this disadvantage is outweighed by a couple factors. The first factor is that we can produce better scores by manual tuning than we can through automation. The second factor is that we have a limited amount of training data available. The data used for training needed to be marked with summary content units. There is a limited amount of such data available. For this reason, even if a mathematically optimal combination of features was found it is doubtfully that such a combination would be optimal on a much larger sample of training data.

There were several general methods used to combine features. All heuristics utilized a *lexical matching* score as a base statistic. A quick test done using the training data determined that a plain lexical matching score produced considerably better results than a lexical matching normalized by the length of the sentence in words. A second experiment was conducted to determine whether matching the full word versus a lemmatized version of the words provided by the semantic analyzer. In this experiment the lemmatized matching produced better results.

The next two features are *sentence type* and *sentence function.* Both of these features provide information about a sentence that pertains in a direct way to the actual content of a sentence. For this reason, it is clear that these features should play a secondary role in the ranking and selection. In heuristics where these features were used, they either provide a slight increase or penalty to the lexical matching score.

Two filtering features, *sentence too long* and *sentence too short*, are used to remove sentences which are not good candidates for summaries. A sentence which is too long will consume too much of the summary space to be useful, and a sentence which is too short is either not a real sentence (for example a sub-heading), or, if it is a complete sentence, it does not provide much useful information. In many ways, not normalizing the lexical match scores creates the necessity for these features. They are also helpful for removing odd strings of text which were not correctly filtered during initial document processing.

The feature that checks is a sentence has a *main clause* was also used to filter out sentences which appeared, based on the semantic analyzer, to be either poorly formed or too long or too complex to flow well within a summary. Sentences lacking a main clause, as identified by the analyzer, were simply filtered out.

The four features (*past, future, perfect* and *progressive*) which involve verb forms were used in two ways. Since they were binary features, it was possible to use them to provide a weighted increase or decreases in the overall score for a sentence. The fact that they are binary features also allows them to be used as a filter to remove certain types of sentences.

The remaining features: *person, mood, grammatical case*, and *grammatical degree* were all added in such a way that they either boost or penalize the lemmatized lexical match score.

3.3.4 Combining formulas

The simplest formula is to use only the lemmatized lexical match feature. This single feature formula also served as a baseline by which to compare other results to. For the 2005 data this produced a total SCU count across all document sets of 107. For the 2006 data the total SCU count was 117. We were not able to normalize the SCU scores for the 2005 data, since the normalizing factors for pyramid evaluation were not available. For 2006, the normalized value (mean-modified SCU score) was about 0.1738. This alone outperforms a large number of systems submitted at DUC.

Because of the lack of the normalizing factors, all tuning was performed using total SCU counts rather than mean modified SCU scores. Due to the limited amount of training

data available, there were in some cases some considerable differences between performance on the 2005 and 2006 data for a given method of combining features. This is further complicated by the fact that all of the topics are different, causing a great variance in results. For our top performing system configuration described later, the standard deviation was approximately 0.14 along side to a mean of approximately 0.21.

Additionally a few features were borrowed from an existing summarization system (Copeck and Szpakowicz, 2005). This existing system had a large number of features. Preliminary experiments indicated that only a few were particularly helpful when combined with the features created by the semantic analyzer. These features were: the number of characters, the number of words, the paragraph number, and the position within the paragraph. A complete list of features is as follows appears in Table 3.12.

A	Lemmatized Lexical Match	K	Verb Form - Perfect
B	Location Correction	L	Verb Form - Progressive
C	Nationality Correction	M	Person
D	Sentence Function	N	Grammatical Mood
E	Sentence Type	O	Grammatical Case
F	Sentence Too Long	P	Grammatical Degree
G	Sentence Too Short	Q	Total Characters
H	Has Main Clause	R	Total Words
I	Verb Form – Past Tense	S	Paragraph Number
J	Verb Form – Future Tense	T	Position in Paragraph

Table 3.12: Final List of Features Used for Heuristics

The first thing we look at is the effect of adding the semantic features individually. This examination was done using the DUC 2005 training data. The main purpose of this was to examine how useful these features are individually so the effect of each one can be determined. These features were combined with the lexical match score using weights and formulas (see Table 3.13). A side-effect is that this may not produce an optimal combination.

Scoring Function	2005 Weighted SCU count
A	263
A+4*B	271
A+5*C	273
A+5*D	300
A+5*E	298
A+0.5*H	267
A+I	240
A-5*J	308
A-K	249
A+5*L	284
A+10*M	285
A+5*N	283
A+O	267
A+P	267

Table 3.13 Adding Features Individually

The development of the above features involved considerable tuning and experimentation. Throughout this process various tests were run using the 2005 weighted SCU count as a measure of success. The design of many of the features was constantly evolving throughout this process. Below, Table 3.14, presents a sample of the SCU scores on a variety of the scoring functions attempted. They have been ordered by the 2006 Mean modified SCU score.

The selections of features and combining formulas listed in Table 3.14 can by no means be considered optimal since, they are derived by manual tuning. These combinations do demonstrate the potential of using this type of semantic features in summarization. When compared with a number of submissions to the DUC 2006 summarization task, this system was able achieve results statistically comparable with some of the top systems and certainly outperformed many of the systems.

These features can also be combined with other types of features derived from various other methods. This is an important feature of a summarization method. Consider that the top-performing system only achieve approximately 20-25% of the SCU count of the average of the manual summarizers. This certainly implies that there is considerable room for improvement in even the best performing systems.

Scoring Function	2005 Weighted SCU count	2006 Weighted SCU count	2006 Mean modified SCU score
10*(A+B+C+D+E)+15*(1/S)+10*(A+B+C+D+(1-F)+(1-G)+20*H+4*M)	334	325	0.2106
10*A+B+C+D+E+(1-F)+(1-G)+20*H+4*I-J-K+2*N+3*M	321	312	0.2011
A+B+C+D+K+L	303	304	0.1946
A+B+C+D+L	278	296	0.1911
10*A+B+C+15*(1/S)	278	296	0.1911
10*A+B+C+D+E+(1-F)+(1-G)+20*H+6*I-3*J+L+2*N+3*M	318	292	0.1904
A+B+C+D+(1-F)+(1-G)+20*H+2*I+2*N	320	297	0.1903
A+B+C+D+K+L+M+N	300	281	0.1828
A	263	280	0.1739
A+B+C	267	278	0.1737
A+B+C+D+J+K+L	252	258	0.1623
A+B+C+D+E+I+J+K+L	214	257	0.1587

Table 3.14: Training and Testing Results for Various Heuristics

3.3.5 How the System Ultimately Works

The final system performs summarization by selecting sentences from the source documents. To do this, all sentences in all documents and topics are parsed using Machinese Semantics. After each sentence has been parsed, a vector of features is created using the semantic information as described above. The sentences are then ranked using the scoring functions described in the previous section.

At this point, we consider the length of each sentences and the maximum allowed length of the summary. In our case this is 250 words, but this is a parameter that can be easily altered to suit particular needs. We can then build the summary adding one sentence at a time from the top of the ranked list. We keep track of how many words each sentence is and subtract from our total available words. When the next sentence on our ranked list does not fit without going over our limit we stop the process. At this point we have a completed summary.

Chapter 4

Results of Experiments

4.1 Machine Learning Trials

In this work, there were two broad approaches to utilizing the Connexor Machinese Semantics software to improve recall in user-oriented multi-document summarization. The first approach was to attempt to use several off-the-shelf machine learning algorithms to increase the density of candidate sentences available to the selection algorithm. Within the raw document set, approximately 10% of sentences contain at least one SCU. Ideally, to achieve a high SCU score, all sentences selected for the final summary would come from these 10% of sentences. In the random case, only 10% of sentences selected would contain at least one SCU. As a result of this, a summarizer must be fairly clever about picking sentences in order to realize a high pyramid evaluation score.

If the set of candidate sentences contained a considerably higher proportion of sentences with at least one SCU, it could be more likely that the final summary would contain a greater proportion of sentences with at least one SCU. If a machine learning algorithm, run on the full set of candidate sentences, selected every source sentence as possibly containing at least on SCU, a precision of 0.1 and a recall of 1.0 would be achieved. To increase the proportion of candidate sentences containing at least one SCU, the precision measure from a ML classifier therefore needs to be increased to a value greater than 0.1.

Given that an extractive multi-document summary only has space for a handful of the total number of sentences available, a large number cannot and will not be used. For this reason, a considerable drop in recall is acceptable without reducing the collection of sentences to the point where there are insufficient sentences remaining to produce a summary. That being said, it

would be undesirable to be left with less than about 5% of the original sentences for summary production since this would sharply reduce the chances of selecting sentences pertinent to the topic. The amount of sentences required to be remaining depends on the summarization problem, with 5% being an approximate minimum. In our problem we have about 50 documents with about 20 sentences each. This gives us about 1000 sentences to select from. Taking 5% of these sentences leaves a set of about 50 sentences for selection. If only about 10% of sentences contain only one SCU we will have only about 5 sentences containing a SCU in our final set. This is certainly only the bare minimum we would require to produce a summary.

This experiment was conducted in three phases. The first phase utilized a feature vector from an existing summarization system (Copeck and Szpakowicz 2005) that did not contain any information from the semantic parser. This experiment investigated the idea of increasing the SCU density independent of the semantic parser. The second phase used only features from the semantic parser as well as lexical matching. This investigated the suitability of the semantic parser for this task when compared with a more common feature vector. The third phase of the tests used all the information available in order to to achieve the best results possible.

A series of four common classifies were used for this test, as well as a voting classifier making use of all four. In all cases except the J48 classifier applied to the complete set of features, achieved reasonable gains in precision. In a few cases the recall was too imprecise to allow for enough sentences to permit the eventual creation of a summary.

Within this result, simple classifiers such as decision trees such as J48 from Weka (Witten and Frank, 2005) worked best. A strong performance was also achieved using the voting meta-classifier in Weka. In the case of voting, the sub-classifiers used were J48, Naïve-Bayes (NB), IB1, and multi-layered perceptron (MLP).

Precision is the most significant measure in the machine learning trials since it corresponds to what proportion of the sentences in the resulting data set contained SCUs. It is also important to ensure that a sufficient number of sentences are selected to produce summaries. A recall of at least approximately 0.05 is generally required in order to achieve this. The choice of this number is arbitrary for this trial. If this method showed promise it would be possible to develop an exact value or a formula for a value based on the size of the initial document set and the size of the summaries. It would vary by summary topic. When considering the precision

measure, the goal must be considered. All precision measures are in comparison to a baseline precision of approximately 0.1. This represents the proportion of sentences in the initial document set which contain SCUs. While compared to other machine learning experiments a precision of for example 0.3 may not seem high, for this work it represents a considerable increase in the density of sentences containing at least one SCU within the set of candidate sentences. Essentially, the probability of selecting a sentence containing a SCU is triple that of the raw set.

Table 4.1 details the results of machine learning on this problem using three feature sets. "All" is the entire set of features we have available, "Existing" refers to the set of features that do not utilize the semantic parser, and "New" refers to the set of features created using the semantic parser as well the base lexical match statistic.

Feature Set	Classifier	Precision	Recall	F
All	J48	0.362	0.070	0.117
All	IB1	0.201	0.155	0.175
All	NB	0.215	0.380	0.275
All	MLP	0.250	0.072	0.112
All	Voting (all above)	0.319	0.099	0.152
Existing	J48	0.329	0.061	0.103
Existing	IB1	0.169	0.162	0.165
Existing	NB	0.206	0.369	0.264
Existing	MLP	0.336	0.048	0.084
Existing	Voting (all above)	0.295	0.120	0.171
New	IB1	0.126	0.230	0.163
New	NB	0.262	0.037	0.065
New	MLP	0.292	0.042	0.073
New	Voting (all above)	0.252	0.039	0.068

Table 4.1: Machine Learning Results

The addition of the new features obtained using the semantic parser, also considerably improved the results obtained by most of the learners. Features obtained using the semantic parser did not perform as well on their own as they did when combined with the features from the existing system.

The results from the second part of the experiments were more varied. Using the new features generated from the output of the semantic parser did provide a moderate improvement in

the weighted count of SCUs selected for the summaries. When using only the features generated from the semantic parser, the J48 classifier produced a trivial classification that is obviously not useful to this task.

On the other hand, using any of the machine learning algorithms as the sole method of selecting sentence for the summaries produced results that were significantly worse than those generated by the baseline lexical matching system. In order to determine what sentences to include in a summary Weka's "output probabilities" were used as the sentence ranking method within this experiment. These corresponded to the score produced for each sentence to determine what side of the binary classification (containing a SCU or not) a sentence should be placed on. A sentence which is more likely to contain a SCU based on the classifier's decision receives a higher score. Basic trials selecting sentences using Weka's output probabilities on the training data produced results that were not even proportional to the results produced by the baseline system. Consequently, there was no reason to pursue this further and formalize the results.

Since the machine learning output probabilities were not useful in ranking sentences, the results of the machine learning classifier were instead used as an additional feature towards the sentence selecting heuristics. While using the machine learning did outperform a baseline lexical matching system, it did not outperform the results generated by using feature combining heuristics developed by configuring the system; the later method produced the maximum SCU score over DUC 2005 data.

4.1.1 Conclusions

Using a selection of off-the-shelf machine learning algorithms, an attempt was made to increase the proportion of sentences containing at least one SCU. It was possible to realize a considerable improvement in the proportion of sentences containing SCUs using a number of the algorithms. Unfortunately, when combined with other methods for selecting sentences for a summary, it does not appear that pre-filtering sentences using machine learners produces a superior output in terms of SCU count. A possible explanation of this is that the machine learning algorithms filtered out mostly sentences which are not likely to be selected using other summarization methods.

The second portion of this experiment looked at using the classifiers as a sole method of selecting sentences for inclusion in summaries. This however produced no useful results and performed worse than primitive methods.

In conclusion, applying machine learning classifiers on the type of data available for this problem as a method of efficiently combining features does not appear to improve results. Developing heuristics manually produced superior recall in summaries based on the measure of summary content units.

4.2 Evaluation of DUC 2006 Summaries

There were two general methods for summary evaluation. The first is manual evaluation. It is perceived to be the most reliable evaluation because it is not subject to the inaccuracies of automatic systems. There is still some variance in evaluation, because not all humans will share the same opinion on the quality of a summary. See section 4.2.3 for more details on evaluator agreement. In an attempt to study this aspect, we manually reevaluated one of the top tanking systems from DUC.

The second method of evaluation is automatic evaluation. In this work we explore two methods: summary content units and ROUGE. Both of these methods a weakness from the fact that automatic systems do not always interpret things correctly and can also be tricked by an eager party. Automatic evaluation does present the opportunity for researchers to evaluate a larger number of summaries cheaply and quickly. This is particularly useful when tuning a system. This is a key property that this work takes advantage of.

4.2.1 Manual Evaluation Results

A manual evaluation was performed on a limited amount of data. It permits an accurate evaluation of how good the system is. It also provides the only opportunity to evaluate the linguistic quality of systems. While no direct attempt was made in this work to improve linguistic quality directly, it is still important to evaluate it to ensure that we have not built a system which performs well on responsiveness measures, but is impractical to use due to major problems with readability. For all manual evaluations, the volunteers reevaluated the DUC submission 23. This submission came from Peking University and the IBM China Research Lab

(Li, Ouyang et al., 2006). This was the top scoring system in the mean-modified SCU score evaluation. A second limitation of the limited amount of data is that it made it impossible to calculate statistical significance in any useful way because the confidence intervals become too wide. Consequently we can only compare the average scores.

4.2.1.1 Responsiveness

In terms of manual responsiveness, the local human evaluators scored system 23 similarly to the way NIST evaluators scored it. The average score for submission 23 was approximately 12.5% higher from the local evaluators than it was from the NIST evaluators. Assuming a similar difference in marking, a NIST score for the baseline would be around 2.9260 and the system with the new semantic features scores about 2.737.

System	Average Score
DUC 2006 System 23 – Local Re-evaluation	3.3750
Baseline Lemmatized Lexical Match	3.2917
New Features – Best Combination	3.0417

Table 4.2: Average Responsiveness Rating

Given the limited number of summary topics a human evaluation can be conducted on, the confidence intervals for this test are very large. That being said, both the baseline system and the new system both appear to fall within the confidence bounds for the top systems at DUC. In terms of this measure the baseline lexical match system performed superiorly compared with many of the systems submitted to the DUC competition (see Appendix E).

4.2.1.2 Grammaticality

The new system was not trained or tuned for grammaticality. The purpose of this evaluation was to ensure that measures taken to improve responsiveness do not work against the grammaticality of the summary.

System	Average Score
DUC 2006 System 23 – Local Re-evaluation	3.9167
New Features – Best Combination	3.8333
Baseline Lemmatized Lexical Match	3.3333

Table 4.3: Average Grammaticality Rating

For the manual grammaticality evaluation, the local evaluators scored submission 23 at about 5.8% less than the NIST evaluators did. A similar difference places the new system at

about 4.056 and the baseline system at about 3.5266. This evaluation had narrower confidence intervals than the responsiveness evaluation. The new system performed in the mid to high range in terms of ranking compared with DUC systems. The new system also outperformed the baseline system by a fair margin.

For this evaluation the baseline system was ahead of some DUC submissions, but unlike the responsiveness evaluation, did serve more as a literal baseline with most systems outperforming it (see Appendix E).

4.2.1.3 Non-Redundancy

System	Average Score
DUC 2006 System 23 – Local Re-evaluation	3.7292
New Features – Best Combination	3.3542
Baseline Lemmatized Lexical Match	3.3333

Table 4.4: Average Non-Redundancy Rating

The non-redundancy evaluation measure was included in the 2005 and 2006 DUC evaluation but was left out of the 2007 DUC evaluation (DUC, 2005; DUC, 2006; DUC, 2007). In the 2006 evaluation, the scores were extremely flat with most of the submissions falling within the confidence intervals of each other.

In this evaluation the local evaluators scored submission 23 about 8.9% lower than the NIST evaluators did. Assuming the same differential the new and baseline scores would be 3.6517 and 3.6290 respectively. This still places both of them last in terms of rank order.

4.2.1.4 Referential clarity

System	Average Score
DUC 2006 System 23 – Local Re-evaluation	3.7708
Baseline Lemmatized Lexical Match	3.5208
New Features – Best Combination	3.3333

Table 4.5: Average Referential Clarity Rating

For referential clarity, there was only a 2.4% difference between the scores provided by the NIST evaluators and the local evaluators. Applying the same correction to the baseline and new systems, 3.4394 for the baseline and 3.2563 for the new system is obtained. This barely

changes the rank order of the systems and the new and baseline systems both scored within the same confidence bounds.

For this measure, the baseline obtained from the simple lexical match system with stop words removed was one of the top performers. This might suggest that there may not be many systems that contain mechanisms to improve this measure.

4.2.1.5 Focus

For the focus evaluation, the NIST evaluators and local evaluators had a fair bit of disagreement. Given the almost 20% difference between the scores, the local evaluation for this topic would likely be very unreliable. A difference of that size would put the baseline and the new system somewhere in the middle of the field. This evaluation was conducted for completeness. Neither system was trained for this measure.

System	Average Score
DUC 2006 System 23 – Local Re-evaluation	3.1667
Baseline Lemmatized Lexical Match	3.0000
New Features – Best Combination	2.9167

Table 4.6: Average Focus Rating

4.2.1.6 Structure and Coherence

For the structure and coherence measure, there was an 8.9% difference between the NIST and local assessors. A similar difference would place the baseline at about 2.3530 and the new system at about 2.2957. This baseline exceeds the scores of approximately half of the systems, including the new configuration, although the difference is not statistically significant.

System	Average Score
DUC 2006 System 23 – Local Re-evaluation	2.8750
Baseline Lemmatized Lexical Match	2.5625
New Features – Best Combination	2.5000

Table 4.7: Average Structure and Coherence Rating

4.2.1.7 Conclusions

An evaluation similar to the one conducted by NIST for the DUC competition as conducted as a means of comparison. As part of this process, the top-performing system from

the 2006 DUC summarization task was re-evaluated along side the systems being manually evaluated. This provided a comparison how different pools of evaluators evaluate different the same summary. Although a full evaluation of all aspects of the summary was conducted, we should remember that the system's sole aim was to improve responsiveness.

Across this evaluation there was somewhat of a mix of results. The system from DUC generally preformed the best. This was expected since it was the top system from DUC. The baseline system and the system using the new features generally remained in the same general area of the rankings. A noticeable observation is that the scores for the simple baseline system were very comparable to, and often exceeded, the scores for many of the submissions at DUC. In terms of the responsiveness measure, which the system was designed to improve, all three systems do appear near the top of the rank order list from DUC.

One of the most noticeable conclusions from manual evaluation is that the opinions of evaluators vary greatly as to the score of a system. Improving this would likely require evaluation measures and structures which are considerably more rigid and detailed. The other obvious problem with manual evaluation is that expectations of summary content can vary considerably depending on the backgrounds of evaluators and their overall expectations for what a summary should contain.

4.2.2 Ranking of Summaries by Humans

Given the limited number of topics to test and evaluate summarization systems on, it is very hard to achieve differences in scores between systems that are statistically significant at any reasonable and common statistical significance threshold. In an attempt to establish firm differences between the three systems that the local evaluators were asked to evaluate, a ranking question was included.

An additional difficulty with human evaluation of summarization is that opinions of what a summary should contain can vary considerably between one person and the next. Additionally, different evaluators can have a varying tolerance for exactness within explanations. For example someone who is an expert in a field can easily become annoyed by small details in facts. In the reverse case, someone casually reading about a topic will often be satisfied with a basic understanding of a topic, and may not be as worried about small details.

	New Features – Best Combination	Baseline Lemmatized Lexical Match	DUC 2006 System 23 - Local Re-evaluation
Average Ranking	2.2708	2.0625	1.6667
Frequency Rank 1	8	19	26
Frequency Rank 2	19	16	12
Frequency Rank 3	21	13	10

Table 4.8: Histogram of Summary Rankings

An example of this surfaced in the local evaluation. One of the evaluators was employed as clinical pharmacist in a psychiatry department of a hospital. This evaluator's opinion of the responsiveness of the ADD/ADHD diagnosis and treatment topic (see Figure 4.1) was considerably different than the others (see Table 4.9).

The topic presented the following questions and information requirements:

Describe ADD/ADHD.

How is it diagnosed?

What kind of treatments are there?

Discuss the controversies surrounding its treatment.

Figure 4.1: ADD/ADHD Topic

:

	Evaluator 1	Evaluator 2	Evaluator 3	Evaluator 4 (Pharmacist)
New Features – Best Combination	3	2	2	2
Baseline Lemmatized Lexical Match	4	4	4	1
DUC 2006 System 23 – Local Re-evaluation	3	2	4	2
Average this topic	3.33	2.67	3.33	1.67
Average all topics	2.44	3.61	3.61	3
Average all other topics	2.25	4	3.5	3.08

Table 4.9: Responsiveness Rankings by Evaluator for ADHD Topic

There was a fair bit of disagreement on evaluation of responsiveness for the ADD/ADHD topic. This is particularly evident in evaluation of the summary from DUC submission 23. However, looking at the average scores each evaluator handed out on this topic, evaluator 4

marked considerably harder that the other evaluators. This was not true in general. Across all topics or all other topics, evaluator 4 was not the hardest evaluator. This evaluator tended to give scores that were in the middle of the other evaluators' scores.

Speaking with this evaluator after the evaluation was completed; a number of complaints were noted regarding the summaries for this topic.

1) There was not a single line that fully explained the features of ADD/ADHD
2) There was no mention of the methods of diagnoses (including definitions from the DSM-IV)
3) The treatments were not discussed in full
4) It contained nothing but controversies
5) Failed to pick up the non-stimulant-based treatments that are now available

Points 1, 2 and 3 generally describe the lack of detail within the summaries when it comes to explaining things. There are likely two main reasons for this problem. The first is the size the summaries. This particular topic asks for 4 items. Even for a human-written summary this only allows for an average of about 62 words per item. This does not allow for much detail. The second reason may be the source documents. Since new articles intended for a general audience were used, only a level of detail that will appeal to a mass audience is likely available.

In terms of the controversies this is again likely due to the source documents. They may not be good enough to pick up finer details more objectively. Controversies might be more likely to make the news more than simple facts.

In terms of point number 5, the news articles used for the creation of these summaries are several years old. As a result of this, very new developments may not be appearing in the news from these dates. This may also explain why some of the more basic facts also do not appear: they are older information and consequently are not news-worthy years later.

The final problem with news is that, to some degree, it is already a summary. News articles which have been written for print media need to fit in a finite, often very limited, space. The development of small commuter newspapers has further increased this need to compress information even further. As a result, most of the information in the article is just the facts that

are news. Background information and more detailed information would have to be left off due to space constraints.

4.2.2.1 Conclusion

Instead of attempting to evaluate summaries on a wide variety of measures, manual evaluators were asked to provide a simple ranking of the summaries. This removes much of the ambiguity between small differences in an evaluation measure. This ended up further demonstrating the differences between manual evaluators. All systems received a number of each rank, although there was some difference in frequency between the systems. It also demonstrates that different topics perform very differently under different system configurations.

4.2.3 Evaluator Agreement

We utilize Cohen's kappa coefficient (Cohen, 1960) as a measure of agreement between evaluators. Since the evaluators worked in two non-overlapping groups, two sets of kappa coefficients were computed.

Manual Evaluation	Group A - Average Kappa	Group B - Average Kappa
Responsiveness	0.0063	-0.0132
Grammaticality	0.0309	0.0552
Non-Redundancy	0.0319	0.0440
Referential clarity	-0.0196	0.1073
Focus	0.1835	-0.0248
Structure and Coherence	0.0892	-0.0217
Ranking of Summaries	0.2027	0.0555

Table 4.10: Kappa Coefficients for Manual Evaluations

The kappa coefficients for almost all evaluations were very low. Landis and Koch, (1977) consider values for Cohen's kappa which are below 0.2 as only "slight" agreement, and values between 0.2 and 0.4 are considered "fair" agreement. Values which are negative indicate that the level of agreement between the evaluators is less than the expected level of agreement for a random assignment of scores in each evaluation.

A possible reason for such low kappa scores is the difficulty of evaluating summaries. The basic problem of summarization is not particularly well defined since different readers will have different expectations of what a summary should contain. Different backgrounds of evaluators can also have a considerable effect on the scores they assign to particular summaries. An evaluator with a background in human language technology might, for example, be more

forgiving of problems with structure due to there understanding of the problem. Likewise, an evaluator with more detailed knowledge of a particular subject matter may become annoyed by inaccuracies within a summary. An evaluator without such knowledge may not be able to recognize such inaccuracies.

4.2.4 Automated Evaluation

Two main automatic evaluations were conducted. These evaluations are perhaps not as accurate as a manual evaluation, but the fact that they are automatic enables evaluation to be run on a much larger set of summaries. For comparison, the automatic evaluation scores for two systems are also presented. These represent the top system and the median system in terms of SCU score. The top submission was produced by Peking University and the IBM China Research Lab (Li, Ouyang et al., 2006). The median system was produced by the University of Michigan (Erkan, 2006).

4.2.4.1 Summary Content Units

Table 4.11 details the SCU-evaluation results for summaries of the 2006 DUC data. The "Mean Modified SCU Score" column is the mean modified SCU score for the 2006 DUC data.

Submission	Mean Modified SCU Score	Standard Deviation	95% C.I. Lower	95% C.I. Upper
DUC 2006 System 23 – Local Re-evaluation (23)	0.24223	0.117221	0.2115775	0.2728825
New Features – Best Combination (bc)	0.210581	0.141362	0.1710068	0.25015504
Baseline Lemmatized Lexical Match (lm)	0.173884	0.121815	0.1368442	0.21092378
DUC 2006 System 23 – Local Re-evaluation (33)	0.182475	0.091749	0.155124	0.20982602

Table 4.11: Mean-Modified SCU Scores for DUC 2006 Data

In terms of statistical significance using a 95% confidence, a couple of t-tests were performed. Using the t-test we conclude that the system using the new features was not statistically superior to the baseline system:

H_0: $mmss_{bc} = mmss_{lm}$

H_1: $mmss_{bc} > mmss_{lm}$

If: $t = \frac{\mu_{bc} - \mu_{lm}}{s/\sqrt{n}} \geq t_{0.95}(n-1)$ then reject H_0

$$t = \frac{0.210581 - 0.173884}{0.141362/4.472135} = 1.160947 < 1.729 = t_{0.95}(19) \Rightarrow acceptH_0$$

In comparing the statistical significance of the system with the new features generated from the semantic parser with the top-performing system (in terms of mean modified SCU score) at DUC 2006 we found the difference between the systems was also not statistically significant at the 95% threshold:

H_0: $mmss_{23} = mmss_{bc}$

H_1: $mmss_{23} > mmss_{bc}$

If $t = \frac{\mu_{23} - \mu_{bc}}{s/\sqrt{n}} \geq t_{0.95}(n-1)$ then reject H_0

$$t = \frac{0.24223 - 0.210581}{0.117221/4.472135} = 1.120745 < 1.729 = t_{0.95}(19) \Rightarrow acceptH_0$$

The system built using the semantic information was not the top-performing system, in terms of mean modified SCU score. The 95% confidence interval for the system with semantic information did however contain the mean modified SCU score for the top-performing system at the 2006 DUC challenge. The lack of a statically significant difference between the systems was most likely due to the lack of data rather than properties of the systems themselves. Many of the differences between in the systems at DUC were very minimal. Even between the best and worst performing systems differences were not large, although they were statistically significant. See Appendix E.

4.2.4.2 ROUGE

A further evaluation was undertaken using the ROUGE system to evaluate the highest achieving systems against both the baseline lexical matching system as well as systems submitted by other DUC participants.

In the ROUGE evaluation, the system which used the features generated from the semantic parser performed better than the baseline system which used lexical matching. The systems with and without semantic information were within the statistical confidence intervals of each other. When comparing summaries generated by DUC participants, the system with semantic features ranked in the middle of the list (see appendix E). The ROUGE score was however closer on average to the ROUGE score of the higher scoring systems than it was to the ROUGE score of the lower scoring systems. It also scored higher than our baseline lexical match system. One possible reason the system did not perform as strongly on the ROUGE measures was that it was in no way trained to score high on ROUGE. Some systems at DUC have used ROUGE as a training measure.

System	Score	95% C.I. Lower	95% C.I. Upper
DUC 2006 System 23 – Local Re-evaluation	0.09300	0.07855	0.10677
DUC 2006 System 33 – Local Re-evaluation	0.08058	0.07066	0.09088
New Features – Best Combination	0.07618	0.06433	0.08844
Baseline Lemmatized Lexical Match	0.07136	0.06007	0.08345

Table 4.12: ROUGE-2 Scores for DUC 2006 Data

System	Score	95% C.I. Lower	95% C.I. Upper
DUC 2006 System 23 – Local Re-evaluation	0.14929	0.13617	0.16166
DUC 2006 System 33 – Local Re-evaluation	0.14174	0.13254	0.15120
New Features – Best Combination	0.12580	0.11470	0.13669
Baseline Lemmatized Lexical Match	0.12434	0.11337	0.13577

Table 4.13: ROUGE-SU4 Scores for DUC 2006 Data

4.2.4.3 Conclusion

In automatic evaluation, the most important aspect is what automatic measure was used to train or tune the system. Many systems presented at DUC were trained using ROUGE since it is the easiest and most readily available automatic evaluation system. Consequently, many of the systems which performed well with ROUGE achieved scores closer to the mean when evaluated using other methods such as mean-modified SCU score or manual evaluation.

In the work presented here, SCU-based evaluation was used for tuning. As expected, this system performed well using SCU evaluation. It did not perform as well using ROUGE and manual evaluation. A notable success is that the system described here achieved an average score of 0.210581 within a 95% confidence interval of the score (0.24223) for the top-performing systems at DUC.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Within previous work on automatic text summarization, a number of different approaches have been attempted. These approaches have produced varying results. However, in all cases, these approaches fell short of the results produced when humans manually undertake the task of text summarization. This tends to imply that there is, in general, considerable room to improve automatic summaries. It may, however, require very advanced methods to realize these improvements. These advanced methods may not be easy or even feasible to develop. It is therefore preferable to look for improvements by adding to existing methods and mechanisms. In this work we added a number of features based on semantic information to standard summarization process.

One particular issue that makes the task very difficult is the fact that it is very difficult to precisely define exactly what should or should not appear in a summary. As a result many methods of evaluation rely on a sample of summaries generated manually. The very nature of needing a sample of summaries for evaluation illustrates the level of uncertainty in the task. This level of uncertainty also extends to evaluation of summaries. In general, a summarization system will perform best on the evaluation metric it was tuned to. This includes both manual and automatic tuning. For this work, the system was tuned using summary content units (SCUs). Consequently, the system achieved its best score relative to the performance of other systems when this evaluation metric was used. In order to complete an evaluation which is both honest and complete, a number of additional evaluations were performed using ROUGE. In these

evaluations, the performance varied, but in general, the performance was not as good as it was in the SCU evaluation.

The second phase of evaluation was performed manually. In this evaluation, the system utilizing information extracted from the semantic parser did marginally increase the average scores. It, however, did not bring the responsiveness scores close to the scores of the top-performing system at DUC. What was however evident was the degree of variability in the expectations of the evaluators. In one situation, the evaluator had considerable knowledge of the subject matter and scored it substantially worse than the other evaluators did. Lack of a clear definition of what a good summary is, along with differences between evaluators makes manually evaluation of summaries very inexact.

The addition of some shallow semantic features to query-based summarization systems did not produce dramatic results. Some small improvements were evident. Given the vast amount of information available from a semantic parse, there is significant potential for this type of information to improve query-based multi-document summarization.

Another area of variability was the variety of topics. Within the manual evaluation phase, evaluators were asked to rank three summaries for each topic in order from best to worse. While the top-performing system was ranked the best more frequently than the other systems, all three systems did appear in all three positions multiple times. This variability in ranking appeared even when evaluating with a limited quantity of topics.

While there were some improvements using features derived from a semantic analyzer, we also reaffirmed several things. The improvement was approximately 21% compared with the lemmatized lexical matching. There were, however, more complex systems that performed within 5% better or worse compared to the system that used the semantic features. The simple lemmatized lexical match accounted for most of the score, with the other features accounting for small improvements. The simple lemmatized lexical match also outperforms a large number of the systems presented at the DUC conference. This demonstrates how little progress has been made on automatic text summarization. The performance of a trivial lexical matching system is comparable to many of the more advanced configurations and strategies.

The use of the semantic analyzer improved responsiveness to the level of other high performing methods. The combinations of heuristics and features we tried did not produce

results which were an improvement over other methods. There has, however, been a demonstration that the types of features made available by the semantic analyzer could continue to improve summaries if the methods of producing optimal feature sets and methods of combining features could be determined.

5.2 Future Work

This work has shown that the use of shallow semantic features in the area text summarization as some potential. In terms of future directions, there are a few places to look. Automatic text summarization research can be categorized into two areas: the production of summaries and the evaluation of summaries. Information obtained from a semantic analyzer has the potential to assist in both.

5.2.1 Improving the Quality of Summaries Produced

Connexor Machinese Semantics produces a very large number of features, structures and other information useful to summarization and natural language processing. This work found uses and extraction processes for a limited number of them. There are few additional features that could be experimented with to improve summaries.

There is a multitude of possible information that could be extracted from a semantic parse tree. Consequently there exists an extensive possibility for the creation of new features, particularly those that dig deeper into the tree-structure.

Additionally, the tree structure produced by the semantic analyzer would allow dependencies between works to be explored. This could be done with dependency pairs or something considerably more elaborate such as graph algorithms. A key advantage of the Machinese Semantics tool is that it provides all of the information in one location. This overcomes the not insignificant problem of combining information coming from a variety of tools.

5.2.2 Improving Automatic Evaluation of Summaries

Evaluation is considerable problem with automatic summarization. While it is possible to utilize manual evaluation to evaluate summaries, this method is impractical for day to day

improvements and modifications to systems. As a result automatic evaluation is generally required. This type of evaluation is still a work in progress.

Using the SCU corpus to perform automatic evaluation does improve the prospects somewhat since it permits the automation of an otherwise manual form of evaluation. It does have the draw-back that it is not able to evaluate all possible sentences within a summary,

The dominating form of fully-automated evaluation is ROUGE. There is potential to either enhance ROUGE measures or develop new ROUGE measures using Machinese Semantics.

5.2.3 Improving the SCU Corpus

There are two additional needs that would assist all research on query-based multi-document summarization. The first is to have additional topics complete with pyramid SCUs. At present it is very difficult to train or tweak either a machine learning classifier or a heuristic as different topics have varying levels of difficulty and many of their characteristics are very different.

The second area for future work is to increase the portion of sentences within topics that are tagged with SCU data. Presently only a small portion of the sentences have any SCU tagging. The reason the others do not is that they were not selected by any system submitted to the DUC competition. As a result, potentially many useful sentences for selection have no value attached to them in these types of tests. It would be a very extensive amount of work to SCU mark entire document sets but it would offer an invaluable pool of information of testing summarization systems.

Currently the collection of SCU tagged sentences exhibits high precision with low recall due to the fact that so few sentences are used in extractive summaries. It might be possible to increase the recall using machine learning algorithms (or other methods) along with various features collected from sources including a semantic analyzer to automatically predict sentences with SCUs. Manual evaluation would then be required to determine the quality of the new sentences chosen to contain SCUs.

Bibliography

Alfonseca, E., Moreno-Sandoval, A., Okumura, M., and Guirao, J. M. (2006). Googling Answers' Models in Question-Focused Summarisation. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 139-143, New York, USA.

Blair-Goldensohn, S., and McKeown, K. (2006). Integrating Rhetorical-Semantic Relation Models for Query-Focused Summarization. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 20-27, New York, USA.

Bosma, W. (2006). Query-Based Extracting: How to Support the Answer? In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 159-163, New York, USA.

Connexor, (2003a). Connexor Machinese Semantics,
http://www.connexor.eu/technology/machinese/machinesesemantics/, Connexor Oy, Helsinki, Finland.

Connexor, (2003b). Connexor Machinese Semantics Manual, Connexor Oy. Helsinki, Finland.

Cremmins, E.T. (1996). The Art of Abstracting. Arlington, Virgina: Information Resources Press.

Cohen, J. (1960). A coefficient of agreement for nominal scales. In *Educational and Psychological Measurement*, 20(1):37–46.

Conroy, J.M., Schlesinger, J.D., O'Leary, D.P., and Goldstein, J. (2006). Back to Basics: CLASSY 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 38-45, New York, USA.

Copeck, T., and Szpakowicz, S. (2004). Vocabulary Usage in Newswire Summaries. In *Proceedings of Text Summarization Workshop, 2004 Conference of the Association of Computational Linguistics*, pages 19-26.

Copeck, T., and Szpakowicz, S. (2005). Leveraging Pyramids. In Proceedings of the Workshop on Automatic Summarization (DUC 2005) at HLT/EMNLP-2005, Vancouver, B.C., Canada.

Copeck, T., Inkpen, D., Kazantseva, A., Kennedy, A., Kipp, D., Nastase, V., and Szpakowicz, S. (2006). Leveraging DUC. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 191-197, New York, USA.

D'Avanzo, E., Frixione, M., and Kuflik, T. (2006). LAKE System at DUC 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 198-201, New York, USA.

Dang H.T. (2005). Overview of DUC 2005. In *Proceedings of the Document Understanding Workshop (DUC 2005 at, HLT/EMNLP-2005,* Vancouver, B.C., Canada.

Doran, W., Dunnion, J., and Carthy, J. (2006). IIRG-UCD at DUC 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 152-158, New York, USA.

DUC (2005). DUC 2005 Guidelines. http://www-nlpir.nist.gov/projects/duc/guidelines/2007.html. National Institute of Standards and Technology (NIST).

DUC (2006a). DUC 2006 Guidelines. http://www-nlpir.nist.gov/projects/duc/guidelines/2006.html. National Institute of Standards and Technology (NIST).

DUC (2006b). D U C 2 0 0 6: Task, Documents, and Measures. http://www-nlpir.nist.gov/projects/duc/duc2006/tasks.html (and linked pages). National Institute of Standards and Technology (NIST).

DUC (2006c) DUC 2006 Linguistic Quality Questions.
http://www-nlpir.nist.gov/projects/duc/duc2006/quality-questions.txt. National Institute of Standards and Technology (NIST).

DUC (2007a). DUC 2007 Guidelines. http://www-nlpir.nist.gov/projects/duc/guidelines/2007.html.. National Institute of Standards and Technology (NIST).

DUC (2007b). DUC 2007 Task, Documents, and Measures:Update Task (Pilot) http://www-nlpir.nist.gov/projects/duc/duc2007/tasks.html#pilot. National Institute of Standards and Technology (NIST).

Edmundson, H.P. (1969). New Methods in Automatic Summarization. Journal of the Association for Computing Machinery, 16(2):264-285, April 1969.

Endres-Niggemeyer, B. (1998). Summarizing Information. Berlin: Springer.

Erkan, G. (2006). Using Biased Random Walks for Focused Summarization. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 186-190. New York, USA.

Favre, B., Bechet, F., Bellot, P., Boudin, F., El-Beze, M., Gillard, L., Torres-Moreno, J.-M., and Lapalme, G. (2006). The LIA-Thales Summarization System at DUC 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 131-138, New York, USA.

Fisher, S., and Roark, B. (2006). Query-Focused Summarization By Supervised Sentence Ranking and Skewed Word Distributions. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 93-100, New York, USA.

Fuentes, M., Rodriguez, H., Turmo, J., and Ferrès, D. (2006). SEM: Semantic-Based Multidocument Summarization at DUC 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 144-151, New York, USA.

Hand, T.F. (1997). A proposal for task-based evaluation of text summarization systems. In *ACL/EACL-97 Workshop on Intelligent Scalable Text Summarization*, pages 31-36, Madrid, Spain.

Harman, D., & Over, P. (2004). The Effects of Human Variation in DUC Summarization Evaluation. In *Proceedings of the ACL-04 Workshop: Text Summarization Branches Out,* pages 10-17, Barcelona, Spain.

Harnly, A., Nenkova, A., Passonneau, R., & Rambow, O. (2005). Automation of summary evaluation by the pyramid method. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005)*. Borovets, Bulgaria.

Howy, E. (2001). Automated text summarization. In *Handbook of Computational Linguistics*, R. Mitkov (ed.), Oxford, U.K. Oxford University Press.

Jagarlamudi. J., Pingali, P., and Varma, V. (2006). Query Independent Sentence Scoring Approach to DUC 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 46-53, New York, USA.

Jarmasz, M. (2003). Roget's thesaurus as a lexical resource for natural language processing. Master's thesis, University of Ottawa.

Jurafsky, D. S. and Martin , J. H. (2000). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition., New Jersey, U.S.A. Prentice Hall.

Kan, M. (2001). Domain-specific informative and indicative summarization for information retrieval, In Proceedings of 2001 Workshop of Text Summarization (DUC), New Orleans, Louisiana USA.

Lacatusu, F., Hickl, A., Roberts, K., Shi, Y., Bensley, J., Rink, B., Wang, P., and Taylor, L. (2006). LCC's GISTexter at DUC 2006: Multi-Strategy Multi-Document Summarization. In *Proceedings of th e Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 54-61, New York, USA.

Landis, R. J. and Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. In *Biometrics*, 33:159–174.

Li, S., Ouyang, Y., Sun, B. and Guo, Z. (2006). Peking University at DUC 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 101-106, New York, USA.

Li, W., Li, B. and Wu., M. (2006). Query Focus Guided Sentence Selection Strategy for DUC 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 33-37, New York, USA.

Lin, C. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In *Proeedings of Text Summarization Branches Out*, Post-Conference Workshop of ACL 2004, pages 74-81, Barcelona, Spain.

Lin, C. (2005). Recall-Oriented Understudy for Gisting Evaluation (ROUGE), http://haydn.isi.edu/ROUGE/.

Lin, C. and Hovy, E. (2003). The Potential and Limitations of Automatic Sentence Extraction for Summarization. In *Proceedings of the HLT-NAACL 03 workshop on Text summarization.* Edmonton, Canada.

Lin, D. (1999). Minipar---a minimalist parser. In Maryland Linguistics Colloquium, University of Maryland, College Park.

Litkowski, K.C. (2006). CL Research Summarization in DUC 2006: An Easier Task, An Easier Method? In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 13-19, New York, USA.

Loos, E., Anderson, S., Day, D., Jordan, P., Wingate, D. (2004). Glossary of linguistic terms http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/, SIL International.

Mani, I., and Maybury M.T. (eds.) (1999). Advances in Automatic Text Summarization. Cambridge, Massachusetts, USA. MIT Press.

Mani, I. (2001). Natural Language Processing – Automatic Summarization. Philadelphia, Pennsylvania, U.S.A. John Benjamins North America.

Daniel Marcu. (2000). The Theory and Practice of Discourse Parsing and Summarization. MIT Press.

Melli, G., Shi, Z., Wang, Y., Liu, Y., Sarkar, A., and Popowich, F. (2006). Description of SQUASH, the SFU Question Answering Summary Handler for the DUC 2006 Summarization Task. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 107-114, New York, USA.

Miller G. (1995). WordNet: A Lexical Database for English. In *Communications of the ACM*, 38(11):39-41.

Mohamed A., and Rajasekaran, S. (2006). Query-Based Summarization Based on Document Graphs. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 159-163, New York, USA.

Molla, D., and Wan, S. (2006). Macquarie University at DUC 2006: Question Answering for Summarisation. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 62-69, New York, USA.

Nenkova, A and Passonneau, R. (2004). Evaluating content selection in summarization: the pyramid method. In *Proceedings of the Workshop on Automatic Summarization (DUC 2004), at HLT/ NAACL-2004*, Boston, Massachusetts, USA.

Nenkova, A. and Passonneau, R. (2005). The new pyramid annotation tool. Retrieved from: http://www1.cs.columbia.edu/~ani/DUC2005/Tool.html

Passonneau R. J., McKeown K., Sigelman S., and Goodkind A. (2006). Applying the Pyramid Method in the 2006 Document Understanding Conference. In *Proceedings of the Document Understanding Workshop (DUC 2006), at HLT/NAACL-2006*, New York USA.

Pinto Molina, M. (1995). "Documentary abstracting: Toward a methodological model". Journal of the American Society of Information Science, 46(3): 225-234.

Sanderson, M. (1994) Stop Word List. Retrieved from: http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words

Santos, E., Mohamed, A. A. & Zhao, Q. (2004) Automatic evaluation of summaries using document graphs. In *Proceedings of the ACL-04,* pages 66-73, Barcelona, Spain.

Schilder, F., and Thomson, B.M. (2006). TLR at DUC 2006: Approximate Tree Similarity and a New Evaluation Regime. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 115-121, New York, USA.

Seki, Y., Aono, M., Eguchi, K., and Kando, N. (2006). Opinion-Focused Summarization and Its Analysis at DUC 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 122-130, New York, USA.

Sjöbergh, J. (2007). Older versions of the ROUGEeval summarization evaluation system were easier to fool. In Information Processing and Management: an International Journal, 43(6):1500-1505, November 2007.

Sleator, D. and Temperley, D. (1993). Parsing English with a link grammar. In Proceedings of the Third International Workshop on Parsing Technologies, pages 277-292.

Spark-Jones, K. (1999). Automatic Summarizing: Factors and Directions. In *Advances in Automatic Text Summarization*, I. Mani and M.T. Maybury (eds.), 1-12. Cambridge, Massachusetts, U.S.A. MIT Press.

Sparck Jones, K (2007). Automatic Summarizing: The state of the art. Information Processing and Management, 43 (2007):1449–1481.

Sparck-Jones, K. and Galliers, J. (1996). Evaluating Natural Language Processing Systems: An Analysis and Review. In *Lecture Notes in Artificial Intelligence 1083*. Springer-Verlag.

TAC (2008). Text Analysis Conference: Call for Participation. http://www.nist.gov/tac/.

Teufel, S. and Moens, M. (1997). Sentence extraction as a classification task. In Mani and Maybury (1997), pages 58-65.

Vanderwende, L., Suzuki, H., and Brockett, C. (2006). Microsoft Research at DUC 2006: Task-Focused Summarization with Sentence Simplification and Lexical Expansion. In *Proceedings of th e Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006. pages 70-78, New York, USA.

Witten I. H. and Frank, E. (2005). "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, U.S.A.

Wu, Y.-C., Tsai, K.-C., Lee, Y.-S., and Yang, J.-C. (2006). Light-Weight Multi-Document Summarization Based on Two-Pass Re-Ranking. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 78-84, New York, USA.

Ye, S., and Chua, T.-S.. (2006). NUS at DUC 2006: Document Concept Lattice for Summarization. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 85-92, New York, USA.

Zajic, D.M., Dorr, B., Lin, J., and Schwartz, R. (2006). Sentence Trimming and Selection: Mixing and Matching. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 179-185, New York, USA.

Zhao, L., Wu, L., and Huang, X. (2006). Fudan University at DUC 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 28-32, New York, USA.

Zhou Q., Sun, L., and Lv, Y. (2006). ISCAS at DUC 2006. In *Proceedings of the Workshop on Automatic Summarization (DUC 2006)*, HLT-NAACL 2006, pages 9-12, New York, USA.

Appendix A

Directions to Human Evaluators

Responsiveness Assessment

You have been given a topic statement and three summaries that are supposed to contribute toward satisfying the information need expressed in the topic statement. Some of the summaries may be more responsive to the topic than others. Your task is to help understand how well each summary responds to the topic.

1. Responsiveness

Read the topic statement and all the associated summaries. Then score each summary according to how responsive it is to the topic.

1. Very Poor	2. Poor	3. Barely Acceptable	4. Good	5. Very Good

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

Linguistic Quality

The linguistic quality questions are targeted to assess how readable and fluent the summaries are, and they measure qualities of the summary that DO NOT involve comparison with a topic. The information content and responsiveness of the summary are measured separately in the "responsiveness" part of the evaluation.

All linguistic quality questions require a certain readability property to be assessed on a five-point scale from "1" to "5", where "5" indicates that the summary is good with the respect to the quality under question, "1" indicates that the summary is bad with respect to the quality stated in the question, and "2" to "4" show the gradation in between. For each question, please try to

assess the quality of the summary only with respect to the property that is described in the question.

2. Grammaticality

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

1. Very Poor	2. Poor	3. Barely Acceptable	4. Good	5. Very Good

3. Non-redundancy

There should be no unnecessary repetition in the summary. Unnecessary repetition might take the form of whole sentences that are repeated, or repeated facts, or the repeated use of a noun or noun phrase (e.g., "Bill Clinton") when a pronoun ("he") would suffice.

1. Very Poor	2. Poor	3. Barely Acceptable	4. Good	5. Very Good

4. Referential Clarity

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

1. Very Poor	2. Poor	3. Barely Acceptable	4. Good	5. Very Good

5. Focus

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

1. Very Poor	2. Poor	3. Barely Acceptable	4. Good	5. Very Good

6. Structure and Coherence

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

1. Very Poor	2. Poor	3. Barely Acceptable	4. Good	5. Very Good

---

Overall Ranking

7. Overall Ranking

For each topic there are three summaries presented. In terms of how you would feel about each of these summaries as a user of the information, rank the three summaries from BEST TO WORST. This can be thought of as your overall satisfaction. In the event two summaries are identical, they can be listed in any order.

For example if B is best and C is worst:

B, A, C

Appendix B

Human Evaluation Orderings

Key
A = New system
B = Baseline lexical match system
C = System 23 from DUC 2006

Group A Human Evaluators

	Topics																	
Evaluator	**601**			**615**			**627**			**631**			**643**			**650**		
A1	A	B	C	A	C	B	B	A	C	B	C	A	C	A	B	C	B	A
A2	A	C	B	B	A	C	B	C	A	C	A	B	C	B	A	A	B	C
A3	B	A	C	B	C	A	C	A	B	C	B	A	A	B	C	A	C	B
A4	B	C	A	C	A	B	C	B	A	A	B	C	A	C	B	B	A	C

Group B Human Evaluators

	Topics																	
Evaluator	**603**			**614**			**617**			**628**			**629**			**647**		
B1	A	B	C	A	C	B	B	A	C	B	C	A	C	A	B	C	B	A
B2	A	C	B	B	A	C	B	C	A	C	A	B	C	B	A	A	B	C
B3	B	A	C	B	C	A	C	A	B	C	B	A	A	B	C	A	C	B
B4	B	C	A	C	A	B	C	B	A	A	B	C	A	C	B	B	A	C

Appendix C

Human Evaluation Topics

Group A Human Evaluators

601 Native American Reservation System - pros and cons

Discuss conditions on American Indian reservations or among Native American communities. Include the benefits and drawbacks of the reservation system. Include legal privileges and problems.

615 Evolution/creationism debate

What are the various perspectives in the U.S. public debate regarding the teaching of evolution, creation science, or intelligent design in public school science classes? What are the key points and counterpoints expressed by people who hold each of those perspectives?

627 International adoption

What are the laws, problems, and issues surrounding international adoption by American families?

631 Crash of the Air France Concorde

Discuss the Concorde jet, its crash in 2000, and aftermaths of this crash.

643 El Nino and La Nina weather condition

Describe the causes and effects of the El Nino and La Nina weather condition. What programs and scientific techniques are in effect to better predict and cope with the conditions?

650 Former President Carter's international activities

Describe former President Carter's international efforts including activities of the Carter Center.

Group B Human Evaluators

603 Wetlands value and protection

Why are wetlands important? Where are they threatened? What steps are being taken to preserve them? What frustrations and setbacks have there been?

614 Quebec independence

Describe developments in the movement for the independence of Quebec from Canada.

617 EgyptAir Flight 990

What caused the crash of EgyptAir Flight 990? Include evidence, theories and speculation.

628 ADD/ADHD diagnosis and treatment

Describe ADD/ADHD. How is it diagnosed? What kind of treatments are there? Discuss the controversies surrounding its treatment.

629 Computer viruses

Identify computer viruses detected worldwide. Include such details as how they are spread, what operating systems they affect, what damage they inflict, their country of origin, and their creators wherever possible.

647 Elian Gonzales custody battle

Describe the custody battle between Cuban and US relatives of the boy Elian Gonzales. Include details about how he came into the custody of his US relatives, the legal and international issues, and the resolution of the situation.

Appendix D

Style Sheet for displaying Connexor Machinese Semantics XML format parses in a web browser

```
<?xml version="1.0"?>
<xsl:stylesheet id="msem" version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<!--XSL Style Sheet is for presenting Machinese Semantics XML output as
<!--feature structures.                                                    -->
<!--The output of this displays correctly in Netscape 7.x or Mozilla 1.x.  -->
<!--It defintely does not display correctly in Internet Explorer.          -->
<!--The following files are required to produce the output correctly:      -->
<!--          msem_left-edge.gif                                          -->
<!--          msem_lower-left-corner.gif                                  -->
<!--          msem_lower-right-corner.gif                                 -->
<!--          msem_right-edge.gif                                         -->
<!--          msem_upper-left-corner.gif                                  -->
<!--          msem_upper-right-corner.gif                                 -->

<!--Written by Darren Kipp, University of Ottawa                          -->
<!--           dkipp@site.uottawa.ca                                      -->


<xsl:output method="html"/>
<xsl:template match="/">
  <html><body bgcolor="#eeeeee"><font size="-1">
        <xsl:apply-templates/>
  </font></body></html>
</xsl:template>

<xsl:template match="analysis">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="fs[@name = 'S']">
  <table border="0" cellpadding="0" cellspacing="0">
    <tr><td><table border="0"><tr><td>
              <xsl:value-of select="@name"/>:
            </td>
            <xsl:if test="value/@id">
              <td><table border="1"><tr><td><xsl:value-of select="value/@id"/>
              </td></tr></table></td>
            </xsl:if>
     </tr></table></td>

      <xsl:if test="@value">
        <td colspan="2"><xsl:value-of select="@value"/></td>
      </xsl:if>

      <xsl:if test="value">
        <xsl:if test="value/fs">
          <td><table border="0" cellpadding="0" cellspacing="0">
              <tr>
                <td background="msem_left-edge.gif">
```

```
                <img src="msem_upper-left-corner.gif" border="0"/>
               </td>
              <td></td>
              <td background="msem_right-edge.gif" align="right">
                <img src="msem_upper-right-corner.gif" border="0"/>
              </td>
             </tr>
             <tr>
              <td background="msem_left-edge.gif">
                <table background="msem_left-edge.gif"><tr><td></td></tr></table>
              </td>
              <td><xsl:apply-templates/></td>
              <td background="msem_right-edge.gif">
                <table background="msem_right-edge.gif"><tr><td></td></tr></table>
              </td>
             </tr>
             <tr>
              <td><img src="msem_lower-left-corner.gif" border="0"/></td>
              <td></td>
              <td align="right"><img src="msem_lower-right-corner.gif" border="0"/>
               </td>
             </tr>
          </table></td>
        </xsl:if>

        <table><tr><td><xsl:value-of select="value/@idref"/></td></tr></table>

      </xsl:if>
    </tr>
  </table>
</xsl:template>

<xsl:template match="fs">
  <tr>
    <td>
      <table border="0"><tr>
          <td><xsl:value-of select="@name"/>:</td>
          <xsl:if test="value/@id">
            <td><table border="1"><tr><td>
                <xsl:value-of select="value/@id"/>
             </td></tr></table></td>
          </xsl:if>
      </tr></table>
    </td>
    <xsl:if test="@value">
      <td colspan="2"><xsl:value-of select="@value"/></td>
    </xsl:if>
    <xsl:if test="value">
      <xsl:if test="value/fs">
        <td>
          <table border="0" cellpadding="0" cellspacing="0">
            <tr>
              <td valign="top" background="msem_left-edge.gif">
                <img src="msem_upper-left-corner.gif" border="0"/>
              </td>
              <td>
              </td>
              <td valign="top"  background="msem_right-edge.gif" align="right">
                <img src="msem_upper-right-corner.gif" border="0"/>
              </td>
            </tr>
            <tr>
              <td background="msem_left-edge.gif">
                <table background="msem_left-edge.gif"><tr><td></td></tr></table>
              </td>
              <td><xsl:apply-templates/></td>
              <td  background="msem_right-edge.gif">
                <table background="msem_right-edge.gif"><tr><td></td></tr></table>
              </td>
            </tr>
            <tr>
```

```
            <td valign="top"><img src="msem_lower-left-corner.gif" border="0"/></td>
            <td></td>
            <td valign="top" align="right">
              <img src="msem_lower-right-corner.gif" border="0"/>
            </td>
          </tr>
        </table>
      </td>
    </xsl:if>
    <table><tr><td><xsl:value-of select="value/@idref"/></td></tr></table>
  </xsl:if>
 </tr>
</xsl:template>

<xsl:template match="value">
 <xsl:if test="fs">
  <table border="0" cellpadding="0" cellspacing="0">
   <xsl:value-of select="fs"/>
   <xsl:apply-templates/>
  </table>
 </xsl:if>
</xsl:template>

</xsl:stylesheet>
```

Appendix E

Results tables from the 2006 Document Understanding Conference

This appendix contains the evaluation tables from the DUC 2006 summarization task. The results for the locally evaluated summaries are included in the tables. Some of the tables include confidence interval overlap ranges showing which summaries are within the same statistical significance range. Since the locally evaluated summaries are from a difference statistical distribution they can not be included in these ranges. Consequently they are simply included in rank order based in the absolute score.

DUC 2006 SCU-Evaluation Results and Confidence Intervals				
Submission	**Mean**	**Standard Deviation**	**95% C.I. Lower**	**95% C.I. Upper**
23	0.242230	0.117221	0.2115775	0.27288250
10	0.240620	0.122167	0.2085648	0.27267524
8	0.213120	0.104147	0.1841257	0.24211430
New Features – Best Combination	0.210581	0.141362	0.1710068	0.25015504
27	0.209190	0.105079	0.1796833	0.23869672
28	0.205445	0.104181	0.1759485	0.23494152
15	0.201535	0.098864	0.1733004	0.22976956
2	0.198800	0.068039	0.1792505	0.21834953
6	0.198590	0.130016	0.1612150	0.23596501
3	0.193902	0.106702	0.1629253	0.22491474
24	0.193525	0.091605	0.1668918	0.22015818
33	0.182475	0.091749	0.155124	0.20982602
5	0.176780	0.087764	0.1502723	0.20328769
19	0.176280	0.10286	0.1451769	0.20738314
Baseline Lemmatized Lexical Match	0.173884	0.121815	0.1368442	0.21092378
14	0.173675	0.080605	0.1491537	0.19819628
32	0.168155	0.095711	0.1386603	0.19764966
22	0.168135	0.09466	0.1389628	0.19730721
29	0.164240	0.09363	0.1351185	0.19336146
25	0.151175	0.09542	0.1205486	0.18180137
18	0.135780	0.096491	0.1035951	0.16796486
17	0.130730	0.069803	0.107142	0.15431797
35	0.127160	0.072795	0.1023303	0.15198974
1	0.113405	0.089124	0.0818563	0.14495367

DUC 2006 ROUGE-2 Scores			
System	**Score**	**95% C.I. Lower**	**95% C.I. Upper**
23	0.09300	0.07855	0.10677
12	0.09088	0.08010	0.10205
8	0.08802	0.07648	0.10013
15	0.08780	0.07568	0.09893
24	0.08780	0.07447	0.10229
31	0.08561	0.07340	0.09855
28	0.08410	0.07213	0.09616
10	0.08393	0.07192	0.09632
33	0.08058	0.07066	0.09088
13	0.08027	0.06802	0.09299
6	0.08000	0.06774	0.09323
27	0.07973	0.06820	0.09149
2	0.07956	0.06894	0.08968
32	0.07766	0.06638	0.08937
5	0.07741	0.06744	0.08688
19	0.07730	0.06425	0.09131
New Features – Best Combination	0.07618	0.06433	0.08844
22	0.07494	0.06405	0.08626
14	0.07306	0.06165	0.08552
3	0.07285	0.06184	0.08595
29	0.07156	0.06100	0.08393
4	0.07144	0.06166	0.08176
Baseline Lemmatized Lexical Match	0.07136	0.06007	0.08345
30	0.06967	0.05767	0.08278
9	0.06943	0.05800	0.08237
34	0.06833	0.05865	0.07968
25	0.06591	0.05475	0.07728
7	0.06563	0.05627	0.07596
20	0.06541	0.05752	0.07363
16	0.06450	0.05370	0.07580
17	0.06208	0.05193	0.07275
21	0.06142	0.04778	0.07577
18	0.06030	0.04944	0.07038
35	0.05702	0.04961	0.06486
1	0.05102	0.04085	0.06168
26	0.04567	0.03722	0.05480
11	0.02658	0.02154	0.03224

DUC 2006 ROUGE-SU4 Scores			
System	**Score**	**95% C.I. Lower**	**95% C.I. Upper**
23	0.14929	0.13617	0.16166
24	0.14917	0.13795	0.16150
12	0.14874	0.13963	0.15767
8	0.14431	0.13620	0.15339
15	0.14430	0.13600	0.15238
10	0.14393	0.13509	0.15348
31	0.14263	0.13194	0.15342
33	0.14174	0.13254	0.15120
28	0.14063	0.13081	0.15010
6	0.13654	0.12610	0.14730
2	0.13620	0.12703	0.14478
13	0.13600	0.12580	0.14608
5	0.13484	0.12710	0.14235
32	0.13345	0.12380	0.14352
27	0.13264	0.12307	0.14256
22	0.13234	0.12274	0.14264
3	0.13225	0.12335	0.14239
19	0.13128	0.12061	0.14354
14	0.13059	0.12135	0.14091
29	0.12761	0.11889	0.13701
30	0.12754	0.11681	0.13931
4	0.12714	0.11914	0.13589
7	0.12603	0.11814	0.13505
New Features – Best Combination	0.12580	0.11470	0.13669
25	0.12450	0.11354	0.13465
20	0.12437	0.11730	0.13116
Baseline Lemmatized Lexical Match	0.12434	0.11337	0.13577
34	0.12333	0.11439	0.13264
16	0.12318	0.11253	0.13472
9	0.12221	0.11105	0.13373
18	0.12001	0.11067	0.12982
21	0.11446	0.09810	0.12865
17	0.11320	0.10203	0.12424
35	0.11266	0.10470	0.11986
1	0.10296	0.09412	0.11203
26	0.10107	0.09264	0.10926
11	0.06307	0.05597	0.07101

DUC 2006 Manual Evaluation: Responsiveness		
System	**Score**	
DUC 2006 System 23 – Local Re-evaluation	3.3750	
Baseline Lemmatized Lexical Match	3.2917	
27	3.0800	A
New Features – Best Combination	3.0417	
23	3.0000	A B
10	2.9400	A B C
12	2.9200	A B C D
24	2.8800	A B C D E
31	2.8600	A B C D E
14	2.8200	A B C D E F
28	2.7800	A B C D E F
5	2.7600	A B C D E F
13	2.7000	A B C D E F
6	2.6200	A B C D E F G
3	2.6000	A B C D E F G
32	2.6000	A B C D E F G
19	2.6000	A B C D E F G
8	2.5800	A B C D E F G
33	2.5800	A B C D E F G
30	2.5800	A B C D E F G
22	2.5600	A B C D E F G
4	2.5400	A B C D E F G
2	2.5400	A B C D E F G
20	2.5200	A B C D E F G
7	2.5000	A B C D E F G
15	2.4800	A B C D E F G
29	2.4400	B C D E F G
35	2.4200	B C D E F G
17	2.3800	C D E F G
9	2.3600	C D E F G
21	2.3600	C D E F G
25	2.3400	C D E F G
18	2.3200	D E F G
16	2.3000	E F G
34	2.2400	F G H
26	2.0600	G H
1	2.0400	G H
11	1.6800	H

DUC 2006 Manual Evaluation: Grammaticality		
System	**Score**	
27	4.6200	A
35	4.5200	A B
22	4.4200	A B C
18	4.4200	A B C
29	4.2200	A B C D
23	4.1600	A B C D E
28	4.0800	A B C D E
13	4.0000	A B C D E F
20	3.9600	A B C D E F
DUC 2006 System 23 – Local Re-evaluation	3.9167	
26	3.8600	B C D E F G
1	3.8400	B C D E F G
New Features – Best Combination	3.8333	
3	3.8200	B C D E F G H
2	3.8000	C D E F G H I
5	3.7400	C D E F G H I
21	3.7200	C D E F G H I J
24	3.6400	D E F G H I J
16	3.6400	D E F G H I J
4	3.6000	D E F G H I J
14	3.5800	D E F G H I J
17	3.5600	D E F G H I J
7	3.5200	D E F G H I J K
30	3.5200	D E F G H I J K
31	3.5000	E F G H I J K
15	3.3400	F G H I J K L
Baseline Lemmatized Lexical Match	3.3333	
9	3.3000	F G H I J K L
25	3.2200	G H I J K L
19	3.2200	G H I J K L
33	3.1600	G H I J K L
10	3.1200	H I J K L
8	3.1000	I J K L
6	3.0200	J K L
34	3.0200	J K L
12	2.8400	K L
32	2.7400	L
11	1.3800	M

DUC 2006 Manual Evaluation: Non-Redundancy		
System	**Score**	
35	4.6600	A
1	4.6400	A B
26	4.5800	A B C
30	4.5600	A B C D
27	4.5000	A B C D
18	4.5000	A B C D
11	4.5000	A B C D
7	4.4800	A B C D
22	4.4600	A B C D
10	4.4200	A B C D E
34	4.4000	A B C D E
5	4.3600	A B C D E F
29	4.3600	A B C D E F
17	4.3600	A B C D E F
4	4.3400	A B C D E F
3	4.3200	A B C D E F
2	4.3000	A B C D E F
14	4.2600	A B C D E F
13	4.2400	A B C D E F
9	4.2000	A B C D E F
33	4.1800	A B C D E F
16	4.1200	A B C D E F
25	4.1000	A B C D E F
20	4.0800	A B C D E F
23	4.0600	A B C D E F
21	4.0400	B C D E F
12	4.0200	C D E F
24	4.0000	C D E F
6	3.9800	C D E F
19	3.9600	D E F
8	3.8400	E F
28	3.8400	E F
15	3.8200	E F
31	3.7800	F
32	3.7600	F
DUC 2006 System 23 – Local Re-evaluation	3.7292	
New Features – Best Combination	3.3542	
Baseline Lemmatized Lexical Match	3.3333	

DUC 2006 Manual Evaluation: Referential Clarity		
System	**Score**	
1	4.7000	A
34	4.0000	A B
23	3.8600	B C
DUC 2006 System 23 – Local Re-evaluation	3.7708	
27	3.7200	B C D
Baseline Lemmatized Lexical Match	3.5208	
21	3.4600	B C D E
28	3.4200	B C D E F
24	3.4200	B C D E F
5	3.4000	B C D E F
2	3.4000	B C D E F
13	3.3800	B C D E F
New Features – Best Combination	3.3333	
18	3.3200	B C D E F G
31	3.2600	C D E F G
30	3.2200	C D E F G
14	3.2200	C D E F G
12	3.2200	C D E F G
33	3.2000	C D E F G
8	3.1600	C D E F G H
4	3.1600	C D E F G H
35	3.1600	C D E F G H
6	3.0800	D E F G H I
9	3.0600	D E F G H I
3	3.0200	D E F G H I
17	3.0000	E F G H I
15	2.9800	E F G H I
16	2.8800	E F G H I
32	2.8400	E F G H I
19	2.8000	E F G H I
25	2.7600	E F G H I J
22	2.7600	E F G H I J
29	2.7200	F G H I J
10	2.6400	G H I J
20	2.4600	H I J K
7	2.3800	I J K
11	2.0600	J K
26	1.9000	K

DUC 2006 Manual Evaluation: Focus		
System	**Score**	
1	4.5600	A
27	4.2800	A B
34	4.1200	A B C
24	3.9400	B C D
31	3.8600	B C D E
5	3.8400	B C D E
21	3.8200	B C D E
4	3.8000	B C D E
33	3.8000	B C D E
23	3.8000	B C D E
2	3.7800	B C D E
13	3.7800	B C D E
28	3.7400	B C D E F
15	3.7400	B C D E F
18	3.7200	B C D E F
22	3.6800	B C D E F
12	3.6600	C D E F
30	3.6200	C D E F
8	3.6000	C D E F
3	3.5800	C D E F
6	3.5200	C D E F
17	3.5200	C D E F
14	3.5200	C D E F
35	3.5000	D E F
16	3.4600	D E F
25	3.4400	D E F
32	3.4200	D E F
9	3.3600	D E F
19	3.3600	D E F
29	3.3400	D E F
20	3.3200	E F
10	3.3200	E F
DUC 2006 System 23 – Local Re-evaluation	3.1667	
7	3.1600	F
Baseline Lemmatized Lexical Match	3.0000	
New Features – Best Combination	2.9167	
26	2.5200	G
11	2.5000	G

DUC 2006 Manual Evaluation: Structure and Coherence		
System	**Score**	
1	4.2200	A
27	3.2800	B
34	3.0800	B C
DUC 2006 System 23 – Local Re-evaluation	2.8750	
18	2.8200	B C D
24	2.8000	B C D E
30	2.7800	B C D E
13	2.7200	B C D E F
23	2.6400	C D E F G
22	2.6400	C D E F G
21	2.5800	C D E F G H
Baseline Lemmatized Lexical Match	2.5625	
33	2.5600	C D E F G H
5	2.5200	C D E F G H
New Features – Best Combination	2.5000	
35	2.5000	C D E F G H
31	2.5000	C D E F G H
4	2.4800	D E F G H
2	2.4800	D E F G H
14	2.4200	D E F G H I
3	2.3000	D E F G H I J
20	2.2800	D E F G H I J
28	2.2600	D E F G H I J
17	2.2600	D E F G H I J
29	2.2200	E F G H I J
25	2.2200	E F G H I J
15	2.1600	F G H I J
6	2.1400	F G H I J
16	2.1200	G H I J
9	2.1000	G H I J
7	2.0800	G H I J K
8	2.0600	G H I J K
19	2.0600	G H I J K
12	2.0400	H I J K
32	1.8400	I J K
10	1.8000	J K
26	1.5000	K L
11	1.1600	L