SENTENCE ORDERING FOR MULTI-DOCUMENT SUMMARIZATION
IN RESPONSE TO MULTIPLE QUERIES

by

Yang-Wendy Wang
B.Sc, Northeastern University, 2002

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE in the School of Computing Science

© Yang-Wendy Wang 2006
SIMON FRASER UNIVERSITY
Fall 2006

All rights reserved. This work may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author.
Name: Yang-Wendy Wang
Degree: Master of Science
Title of thesis: Sentence Ordering for Multi-document Summarization in Response to Multiple Queries

Examining Committee: Dr. Ke Wang, Professor, Computing Science
Simon Fraser University
Chair

Dr. Fred Popowich, Professor, Computing Science
Simon Fraser University
Senior Supervisor

Dr. Anoop Sarkar, Professor, Computing Science
Simon Fraser University
Supervisor

Dr. Veronica Dahl, Professor, Computing Science
Simon Fraser University
SFU Examiner

Date Approved:
DECLARATION OF PARTIAL COPYRIGHT LICENCE

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection, and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada
Abstract

The growing access to large amounts of text data opens more opportunities in information processing. Given a list of complex questions and a set of relevant documents, the task of producing an informative and coherent summary of those documents in response to the questions has attracted a great deal of attention recently. However, the problem of organizing information for summarization so that the generated summary is coherent has received relatively little attention. Several approaches have been proposed for sentence ordering in single-document and generic multiple-document summarization, but no single method has been found to address sentence ordering in query-based summarization.

In this thesis, we propose and implement an algorithm that combines constraints from query order and topical relatedness in human produced summaries of multiple documents in response to multiple questions. To test the effectiveness of the constraints, we construct a new query-based corpus from the human produced summaries for the Document Understanding Conference (DUC) 2006 evaluation. We then conduct experiments, using an automatic evaluation method based on Kendall's $\tau$, to evaluate and compare the effectiveness of our approaches to others. Our results show that both query order and topical relatedness improve the ordering performance when compared to a baseline method, and a combination of these two constraints achieves even better results.

Key words: multi-document summarization; sentence ordering; queries; coherence.
To my family
Acknowledgments

My thanks go to all the people whose efforts have allowed me to get to this point. I am indebted to you, and to show my appreciation, I will try my best to carry forward your spirit of giving.

I would like to express my deep gratitude to my senior supervisor Dr. Fred Popowich. I am grateful for his enthusiasm and inspiring me to work in this fascinating field of Natural Language Processing. His support, understanding and patience make him an excellent mentor and adviser.

I would like to thank my supervisor Anoop Sarkar for sharing me with his expertise in natural language processing through insightful comments and fruitful discussions. My gratitude also goes to Dr. Veronica Dahl for serving on my examining committee. I thank her for her precious time reviewing my work and encouragement along the way. I am very grateful to Gabor Melli and Zhongmin Shi for giving me inspiring advices and support throughout the project as well as the thesis work. I really enjoyed the time working together with them. I also want to thank Wolfgang Haas for proofreading my thesis.

I thank all the members in the natural language lab and my other friends for their enthusiasm and continuous support. A particular acknowledgement goes to Dana Brand, Lei Duan, Chris Demwell, Gholamreza Haffari, CongCong Jiang, Michael Letourneau, Yudong Liu, Flavia Moser, Mehdi Mostafavi-Kashani, Javier Thaine, Kun Wu, Carl Zhang. I am very grateful to my dear friend Irene Pekerskaya for all her understanding and encouragement that helped me overcome the difficulties on the way.

Last but not least, I am very grateful to my family, especially my grandparents and my parents for their endless love, support and encouragement. They have me stay in good spirits and pursue my dream.
## Contents

Approval ............................................................. ii  
Abstract .................................................................. iii  
Dedication ................................................................ iv  
Acknowledgments ..................................................... v  
Contents ................................................................... vi  
List of Tables .......................................................... ix  
List of Figures .......................................................... x  

1 Introduction ........................................................... 1  
1.1 Motivation ............................................................. 3  
1.2 Approach ............................................................... 5  
1.3 Contribution .......................................................... 6  
1.4 Thesis Outline ......................................................... 7  

2 Literature Review ..................................................... 8  
2.1 Barzilay et al. .......................................................... 9  
2.1.1 Approach ........................................................... 9  
2.1.2 Data and Evaluation .............................................. 10  
2.2 Lapata ................................................................. 10  
2.2.1 Approach ........................................................... 10  
2.2.2 Data and Evaluation .............................................. 11
List of Tables

2.1 Literature Review .................................................. 19

5.1 The example of the original sentence order in the summary and system generated order ........................................... 51

5.2 Kendall’s τ comparison between different algorithms ........................................... 56

5.3 The Paired Sample Test for different algorithms ....................... 57
List of Figures

1.1 Summary Answer A: Extracted summary without sentence ordering ....... 4
1.2 Summary Answer B (Extracted Summary after sentence ordering) ....... 5

3.1 The Overview of SQuASH Architecture [43]. ............................... 22
3.2 An example of output from the Annotator. ................................. 24
3.3 The correlation between content responsiveness and structure & coherence score for the summary ........................................... 31
3.4 The correlation between content responsiveness and structure & coherence score for the summary when content responsive score equals 1(Content Responsiveness I) and 2(Content Responsiveness II) ......................... 32
3.5 The correlation between content responsiveness and structure & coherence score for the summary when content responsive score equals 3(Content Responsiveness III) and 4(Content Responsiveness IV) .......................... 33
3.6 The correlation between content responsiveness and structure & coherence score for the summary when content responsive score equals 5(Content Responsiveness V) .................................................. 34

4.1 Type A Summary ........................................................................ 42
4.2 Type B Summary ........................................................................ 43

5.1 Comparison between different algorithms ................................... 54
Chapter 1

Introduction

We truly live in an information age. It has never been easier to access and share information. Instead of spending a day at the library we can now accomplish as much and more with a couple of hours of surfing the Web. Search engines and simple question answering systems have been developed to help us find information. Based on ComScore figures\(^1\) in the United States in March 2006, 213 million web searches happen every day, which is 148 thousand searches every minute. But what does the future hold for our interaction with search engines? Instead of the currently retrieved list of documents in response to simple key word searches, it will be useful if we could have a more concise answer with respect to a more complex question like "What health-related problems have been identified with working with computers?" Some web search companies now provide such services like Google Answers and Ask Yahoo, but they rely on human expertise in online search to help provide the answers. Our goal is to allow computers do this job of those professional information analysts. This thesis provides a small step towards this ambitious goal.

The goal of simulating the professional information analyst gave birth to query relevant summarization research, which has attracted a great deal of attention from both Question Answering (QA) and Multi-document Summarization (MDS) research communities in recent years [21]. A synergy exists between text summarization and question answering systems. Summarization provides a way of extracting and fusing the relevant information from different document sources in response to complex questions. Question answering provides

\(^1\)http://searchenginewatch.com/reports/article.php/2156431
a means for focus in query-oriented summarization. The boundaries between these two re-
search communities now are beginning to blur. The focus of this thesis will be more on the
summarization side. We ignore the information retrieval phase for question answering, and
assume we already get the relevant documents for the questions.

Most current query based summarization systems follow what is called the extractive
approach which involves two main stages: content extraction and text representation.

In the first phase, the system processes the multiple documents and extracts the im-
portant content that answers the questions. A wide range of statistical and heuristic based
methods have been tried [20, 22, 15]. Evaluations show that adding a query analysis com-
ponent to the extraction phase greatly improves the system performance [13, 15].

In the second phase, the focus shifts to the creation of a coherent, understandable
“answer” that responds to the original questions based on extracted information. Most
often, the extractive summaries produced from multiple source documents suffer from an
array of problems with respect to text coherence and readability, like dangling references,
irrelevant context cue information, etc. Many approaches [24, 28, 54, 29] have been proposed
to deal with the problems, including co-reference resolution, temporal information recovery
and removal of contextual phrases by sentence compression. But after these post processing
steps, even if each individual sentence might be interpretable in isolation, it still does not
mean that sentences gathered from different sources as a whole will be easy to understand.
Interdependence between sentences will greatly affect readers’ understanding. Therefore,
it is important to consider sentence ordering of extracted sentences in order to reconstruct
discourse structure in a summary.

In this thesis, we propose a query-oriented method to target the task of sentence ordering
for multiple documents in response to multiple questions. Many approaches have been per-
cformed on sentence ordering in single document summarization and generic multi-document
summarization [4, 6, 49]. But there does not appear to be any sentence ordering method
that addresses query based summarization specifically.

The motivation for the study of sentence ordering to answer multiple queries was derived
from the development of the SQuASH(Simon Fraser University Question Answering Sum-
marization Handler) system [42, 43], during the Document Understanding Conference(DUC)
competition organized by the National Institute of Standards and Technology(NIST).
1.1 Motivation

The task in DUC 2005/2006 was to model a real-world complex question answering problem based on query-based multi-document summarization. It requires synthesizing a supplied set of 25 news articles into a concise, well-organized and fluent answer to a need for information that could not be satisfied by just stating a name, date, quantity, etc.

The SQuASH [42, 43] system follows the extractive approach and it has three main components: the Annotator module, the Extractor module and the Editor module. The system starts off by annotating the documents and the question text with the Annotator. These annotations are then fed to the next two summarization modules. The Extractor focuses on sentence selection. The Editor then focuses on linguistic readability by doing sentence compression and sentence ordering.

Two types of evaluation are performed on DUC systems. An automatic evaluation method called Recall-Oriented Understudy for Gisting Evaluation (ROUGE)\(^3\) is used to judge the content salience. The linguistic well-formedness of each submitted summary is manually evaluated by NIST based on its grammaticality, non-redundancy, referential clarity, focus, structure and coherence. In DUC 2006, both content responsiveness and overall responsiveness are evaluated manually. Based on our analysis of the DUC evaluation results, some summaries get very high ROUGE score and content responsiveness for content selection, but their overall responsiveness and linguistic qualities are very poor. The cause of this poor performance includes factors like unresolved anaphors, irrelevant information, grammar errors etc. However we also find that even when individual sentences stand well in isolation, if they are poorly ordered in the summary, then the summary is generally incoherent.

A summary example from DUC 2006, consisting of its id number, title and narratives is provided below, with the extracted summary answers with sentence ordering and without sentence ordering shown in Figure 1.1 and Figure 1.2 respectively. Note that we have introduced sentence numbers into the summaries to facilitate our discussions below.

- **num:**D625G

- **title:**types of diseases in Kenya

---


\(^3\)http://www.isi.edu/~cyl/ROUGE/
1) AIDS is expected to reduce Kenya's gross domestic product by 14.5 percent by the year 2005, according to a latest analysis released by the Ministry of Planning and National Development of Kenya available here. 2) One out of 10 children in Kenya dies of preventable diseases the East African Standard newspaper reported. 3) The Kenyan Department of Public Health will install condom dispensing slots in all public places in Isiolo district in an effort to fight the spread of AIDS and sexually transmitted diseases. 4) Cases of heart attacks in Kenya have increased by almost tenfold a Kenya cardiologist. 5) Almost 80 percent of hospital beds in Kenya's government hospitals are occupied by AIDS patients. 6) The Kenyan government is reviewing its public health act in order to meet the challenges posed by the HIV/AIDS pandemic, the People Daily said. 7) It has established that the used items were causing diseases ranging from leprosy, anthrax and other fungal diseases. 8) Water-borne diseases such as typhoid, cholera and diarrhea swept across the region, claiming scores of lives. 9) The ministers from Kenya, Ethiopia, Djibouti, Uganda, Eritrea and the Sudan will assess the magnitude of tuberculosis, malaria and diarrhea diseases in their respective countries. 10) Research by the National AIDS and Sexually Transmitted Diseases Control indicates that more than 1.9 million adults and 100,000 children in Kenya are infected with HIV/AIDS. 11) Twenty-two people have died of starvation in Fafl area of Garissa District in Kenya's North Eastern Province according to six members of parliament from the province.

Figure 1.1: Summary Answer A: Extracted summary without sentence ordering

- narr:What are the most prevalent diseases in Kenya and and how are they affecting the population? What is being done to combat them?

When comparing summary answers without ordering (Figure 1.1) and one after ordering (Figure 1.2), a summary with ordering reads more fluently, while one without ordering contains abrupt switches of topics. For instance, in Figure 1.1, sentences (1, 5, and 10) about AIDS are split, and appear too far apart in the summary. Grouping them together, as shown in Figure 1.2 increases the readability of the summary.

Another phenomenon in Figure 1.1 is that sentences (3, 6) that answer the second question “What is being done to combat them?” are put before sentences that answer the first question “What are the most prevalent diseases in Kenya and how are they affecting the population?”. This disrupts the logical structure of the answer that a reader expects. The summary answer in Figure 1.2 has a sentence order based on the question sequence. For instance, sentences (7, 3) that answer the second question are shifted to the end of the summary.

We also find that a good clustering of related sentences is very important. For example, in Figure 1.2, some sentences (2, 4, 8, and 11) stand well alone and they do not have a direct cohesion transition with any other neighboring sentences, so different ways of ordering them will create equally readable summaries.
CHAPTER 1. INTRODUCTION

10) Research by the National AIDS and Sexually Transmitted Diseases Control indicates that more than 1.9 million adults and 100,000 children in Kenya are infected with HIV/AIDS. 5) Almost 80 percent of hospital beds in Kenya's government hospitals are occupied by AIDS patients. 1) AIDS is expected to reduce Kenya's gross domestic product by 14.5 percent by the year 2005, according to a latest analysis released by the Ministry of Planning and National Development of Kenya available here. 2) One out of 10 children in Kenya dies of preventable diseases the East African Standard newspaper reported. 4) Cases of heart attacks in Kenya have increased by almost tenfold a Kenya cardiologist. 8) Water-borne diseases such as typhoid, cholera and diarrhea swept across the region, claiming scores of lives. 9) The ministers from Kenya, Ethiopia, Djibouti, Uganda, Eritrea and the Sudan will assess the magnitude of tuberculosis, malaria and diarrhea diseases in their respective countries. 11) Twenty-two people have died of starvation in Fafi area of Garissa District in Kenya's Northern Eastern Province according to six members of parliament from the province. 7) It has established that the used items were causing diseases ranging from leprosy, anthrax and other fungal diseases. 6) The Kenyan government is reviewing its public health act in order to meet the challenges posed by the HIV/AIDS pandemic, the People Daily said. 3) The Kenyan Department of Public Health will install condom dispensing slots in all public places in Isiolo district in an effort to fight the spread of AIDS and sexually transmitted diseases.

Figure 1.2: Summary Answer B (Extracted Summary after sentence ordering)

From the above example, we see that the issue of ordering can affect the overall quality of a summary. In the next session, we will introduce our approach for solving the sentence ordering problem in multi-document summarization in response to multiple questions.

1.2 Approach

Information ordering, which determines the sequence in which to present a set of pre-selected information-bearing items, is a critical task both for text summarization and natural language generation. Several approaches have been taken in solving the information ordering task in multiple document summarization, all of which follow the assumption that the summary structure also follows the structure of the original document set, since multi-document summary captures the main contents among the document clusters. As for query-based summaries, the extracted sentences do not necessarily represent the central meaning of the document set. Meanwhile, readers of query-based summaries often would like to see the answer in the form they want [21]. One simple and direct way to satisfy this request is to structure answers into the cluster for each question and have these questions as the summary skeleton.

The ordering constraints that have been used so far are chronological information [4, 49,
9], topical relatedness [4, 6, 49, 9], majority sentence position in the original article [4, 6], precedence [49, 9] and succession [9].

In this thesis, we propose two ordering constraints to arrange sentences for multiple-document summarization in response to multiple questions. We first group the sentences into question clusters based on the query order constraint and order them based on query structure. Then within each query group, sentences are ordered based on the local coherence calculation between sentences. We use a WordNet based lexical similarity measure to calculate the cohesion between two different sentences on the sentence-level.

Evaluation is another important issue in this thesis. Investigating information ordering extensively by employing human informants in psycholinguistic experiments is often unfeasible [6]. Recent empirical work [30] [49] [6] on information ordering evaluation all adopt automatic methods. Here, we share the assumption that a text production method can be evaluated, albeit approximately, by automatically comparing its output with human-defined solutions provided in a corpus of texts each of which is viewed as a “gold standard” [26]. We evaluate our ordering performance on human written summaries for the DUC2005 evaluation using a current automatic evaluation measure, Kendall’s \( \tau \).

1.3 Contribution

This thesis describes a sentence ordering algorithm for multiple document summarization in response to multiple queries. It is the first ordering algorithm that was proposed specially for query based summarization.

The main contributions in this thesis are the following:

- The construction of the first data corpus that contains human written summaries from DUC evaluation for query based summarization.

- An introduction of a novel ordering constraint that uses query structure to determine the summary structure by clustering the summary sentences into different groups that answer different questions.

- The design of a new similarity function that uses WordNet to incorporate the semantics similarities across sentences to calculate the lexical coherence between two sentences.

- The design of the experiment and the adaptation of the automatic evaluation method
(Kendall's $\tau$) to evaluate the performance of our system generated summary compared with human summary and the baseline.

- The successful application of sentence ordering algorithm for documents in any domain.

- An empirical demonstration that the algorithms performs significantly better than the baseline.

1.4 Thesis Outline

The reminder of this thesis is organized as follows. In chapter 2, we give a literature review of the works in sentence ordering for multi-document summarization in terms of algorithm design, applied datasets, the evaluation methods and applicability to our ordering task in response to multiple queries. In chapter 3, we give the overview of SQuASH system to show how sentence ordering fits in the whole picture and its related evaluation issues. This provides a good context to the whole thesis. Chapter 4 introduces and analyzes a query-based sentence ordering algorithm for multiple document summarization in response to multiple queries. In chapter 5, we discuss the evaluation of the algorithm and its individual ordering constraint using an automatic evaluation measure based on Kendall's $\tau$ method. Finally, in chapter 6, we summarize the findings of the thesis and provide suggestions for future work.
Chapter 2

Literature Review

Although text summarization has been studied since 1958 [33], most efforts have relied on extracting the salient information for the summary. The problem of how to structure the selected information to form a fluent summary has received very little attention until recently. In single document summarization, summary sentences are typically arranged in the same order as they were in the original full document, although it was found that human summarizers do sometimes change the original order [25]. In multi-document summarization, sentences are selected from multiple documents and no complete ordering from a single document is available, so most common approaches involve ordering by the original article publishing time or ordering sentences based on their content importance score from the extraction stage.

Information ordering is also a critical task for natural language generation and has been extensively investigated [41, 11, 52] by the generation community. Sentence ordering for text generation was mostly studied in a domain dependent framework [12], where a priori ordering strategies can be identified through corpus analysis. Since our thesis focuses on sentence ordering in the context of summarization for the general news domain, we will not give the review of the sentence ordering work in text generation research.

Although there does not appear to be any sentence ordering method that addresses query-based summarization specifically, many approaches have been proposed on sentence ordering in generic multi-document summarization. We will review a number of approaches [4, 30, 6, 5, 49, 9] in the following sections in terms of algorithm design, applied datasets and the evaluation methods. In all cases, the discussion will focus not so much on the details of the work itself but rather on its applicability to our ordering task in response to multiple
queries.

2.1 Barzilay et al.

2.1.1 Approach

The first systematic research on sentence ordering was done by Barzilay, et al [4]. They provided a corpus based methodology to study ordering and conducted experiments which show that sentence ordering significantly affects the reader's comprehension. They also evaluated two ordering strategies: majority ordering which orders sentences by their most frequent orders across input documents; chronological ordering which orders sentences by their original article's publishing time. They then introduced an augmented chronological ordering with topical relatedness information that achieves the best results. The augmented strategy used majority and chronological constraints to define the pair wise relations between sentences. Barzilay then identified the final order of sentences by finding a maximal weighted path in a precedence graph [10].

Majority ordering is critically linked to the level of similarity of information organization across the input texts. In the news genre for query-based summarization task, articles often come from different sources and provide different aspects of answers to the questions, so there is not a high level of similarity across texts. Chronological ordering could produce good results when the information is event-based, and therefore, is temporally sequenced. However, with query-based summarization, questions often concern states or properties and thus chronological information is not very useful.

A very important observation from the corpus analysis by Barzilay is that although there are many acceptable orderings given one set of sentences, topical related sentences always share an adjacency relation. She also points out that the notion of grouping topically related sentences is known as cohesion. As defined by Hasan (1984), cohesion is the device for “sticking together” different parts of the text. Good orderings are cohesive; this is what makes the summary readable. This approach requires a robust segmentation algorithm to identify themes which are clusters of similar sentences across different documents. Barzilay approximates theme segmentation by calculating the proportion of the number of sentence pairs which appear in the same text and same segment in the original text over the number of sentence pairs appearing in the same text. In this thesis, such themes are not identified, as no original article information is available, but we follow their insights and treat topical
relatedness as one of important criteria for choosing the neighboring sentences. We will discuss our approach in detail in Chapter 5.

2.1.2 Data and Evaluation

Barzilay et al. collected 25 sets of articles for their experiment and evaluation. Each set consisted of two to three news articles reporting the same event. The extracted sentences for the summary were manually selected, simulating MULTIGEN. The average summary length is 8.8 sentences. Among them, 10 summaries were given another 9 alternative orderings for each set for the study of patterns of summary ordering.

To evaluate different strategies, they ask human judges to manually rank each summary as Poor, Fair, or Good, which are defined as follows.

- **Poor**: Readability would be significantly improved by reordering its sentences.
- **Fair**: A summary makes sense, but reordering of some sentences can yield a better readability.
- **Good**: A summary which cannot be further improved by any sentence reordering.

To assess the significance of improvement, they use the Fisher-exact test (p-value).

Manual evaluation is more reliable than automatic evaluation if inter-human agreement is higher than a certain threshold. But it is often very expensive to construct and results can not be reproduced.

2.2 Lapata

2.2.1 Approach

An unsupervised probabilistic model has been suggested by Lapata for text structuring that learns ordering constraints from sentences represented by a set of lexical and structural features. It assumes the probability of any given sentence is determined by its previous sentence and learns the transition probability from one sentence to the next from the BLLIP

\(^1\)The corpus is available at http://www.cs.columbia.edu/~noemie/ordering/

\(^2\)MULTIGEN is part of the Columbia Summarization System. It operates on a set of news articles describing the same event, creating a summary which synthesizes common information across documents.
corpus based on the Cartesian product between two sentences defined using the following features: verbs and their precedent relationships; nouns (entity-based coherence by keeping track of the nouns); and dependencies (structure of sentences). The overall ordering of the sentences in the summary is learned by greedily searching for a maximal weighted path through the graph, in a manner similar to [4]. Based on the experimental results, she finds that entity-based coherence and the verb-noun structure features are significantly better than any other features.

Lapata shows that the lexical and structural information is very important for the ordering task, but learning those interesting lexical features requires a large corpus. She uses the BLLIP corpus which contains 30 million words. The query based summarization corpus we are using in the thesis is comparatively very small, so the probability calculation for feature learning will encounter the sparse data problem. Although we could train our model on a different and bigger corpus and then test on our own corpus, we are more interested in exploring the relations between queries and sentences in the summary genre. Such query and summary information are not provided by the BLLIP corpus.

2.2.2 Data and Evaluation

Lapata presented an experimental setting which employs the distance between two orderings to estimate automatically how close a sentence ordering produced by her probabilistic model stands in comparison to orderings provided by several human judges. The task is to recover the originally human authored text.

Lapata [30] is the first person who attempted to evaluate sentence ordering in text summarization quantitatively using an automatic performance measure. The automatic evaluation metric she proposed is Kendall's $\tau$. Given an unordered set of sentences and two possible orderings, $\tau$ is used to calculate the distance between them. More details about $\tau$ measure and its comparison will be provided in chapter 5.

The model is trained and tested on the BLLIP corpus, which contains a complete, Treebank-style, parsing of the three-year Wall Street Journal (WSJ) collection, approximately 30 million words as mentioned earlier. The average article length is 15.3 sentences. The model generated orders are compared with the original text order using the $\tau$ evaluation method. A random order is generated as the baseline for the lower bound of the $\tau$ value. The upper bound of the $\tau$ value is determined by conducting an experiment to compare the
model's performance with humans, where human subjects were invited to order the scrambled sentences of 12 texts from the test set and create an additional 33 orderings per text. This method of using human agreement as the upper bound in a corpus-based evaluation provides an alternative to the view of the corpus text as an absolute gold standard.

She also tests her methods on the multi-document summarization corpus that Barzilay et al. [4] has created and achieved competitive results. She performed Post-hoc Tukey tests\(^3\) to examine the significant differences among the different features and between models on above experiments.

Automatic evaluation enables us to assess the importance of particular feature combination easily. Kendall's \(\tau\) seems particularly appropriate for our sentence ordering task. It is sensitive to the fact that some sentences may be always ordered next to each other even though their absolute orders might differ. It also penalizes inverse rankings, which seems appropriate given that flipping the sentences that answer the second question with sentences that answer first question would seriously disrupt coherence. In our research, we adopt their evaluation metric and use \(\tau\) for the ordering score.

\[2.3\] Barzilay and Lee

\[2.3.1\] Approach

Barzilay and Lee [6] have proposed domain-specific content models to represent topics and topic transitions for sentence ordering. They learn the content structure directly from unannotated texts via analysis of word distribution patterns based on the idea that "various types of [word] recurrence patterns seem to characterize various types of discourse" [55]. The content models are Hidden Markov Models (HMMs) wherein states correspond to types of information characteristic to the domain of interest, and state transitions capture possible information-presentation orderings within that domain.

The success of the distributional approach depends on the existence of recurrent patterns. Domain specific texts tend to exhibit high similarity, while in our task, the news articles came from different domains, which lack the recurrent property. But we follow their assumption that formulaic text structure facilitates readers' comprehension [2]. Instead of content patterns, we propose a method using question order to capture the overall text

\(^3\)http://www.uwsp.edu/psych/cw/statmanual/posthocs.html
structure. Barzilay and Lee capture the topic clusters via complete-link clustering and measure sentence similarity with the cosine metric using word bigrams as features. In our case, we create topic clustering based on the semantic similarity between question and sentences. The topic clusters are then ordered based on the order of the questions.

2.3.2 Data and Evaluation

To evaluate their content model, Barzilay and Lee created corpora from five domains: earthquakes, clashes between armies and rebel groups, drug-related criminal offenses, financial reports and summaries of aviation accidents. The average length of articles is 12 sentences. The corpora are domain specific and no queries are involved, so we can not use them for our task.

Three measures are given to evaluate the system performance. The first measure is the average original sentence order (OSO) rank. Since the content models compute the probability of generating each of the permutations of a given document’s sentences, it is easy to get the OSO rank among all the alternative orderings. The best possible rank is 0 and the worst is $N! - 1$, where $N$ is the number of sentences in the document. To compare their system with Lapata’s [30], they also report the OSO prediction rate, which measures the percentage of test cases in which the model gives highest probability to the OSO from among all possible permutations, as they expect that a good content model should predict the OSO a fair fraction of the time. To assess the quality of the predicted orderings themselves, they follow Lapata’s approach in employing Kendall’s $\tau$ [30] as discussed in 2.2.2.

Barzilay and Lee also compute the learning curve for different domains and show that the model performance improves as the size of training sets increases. But they do not report any statistical tests to verify that the observed differences are significant.

Our ordering method is deterministic and no alternative ordering is produced, so the rank metrics cannot be computed for our task. To follow the convention, we chose the Kendall’s $\tau$ measure as our evaluation measure.

2.4 Bollegala, Okazaki and Ishizuka

2.4.1 Approach

Different criteria (e.g.: chronology, topical-closeness, etc.) have been shown to be effective for sentence ordering, but they are investigated separately or linearly. Bollegala, Okazaki and Ishizuka [9] provide a novel supervised learning framework to integrate different criteria. They also propose two new criteria precedence and succession developed from their previous work [49]. A fundamental assumption for the precedence criteria is that each sentence in newspaper articles is written on the basis that pre-suppositional information should be transferred to the reader before the sentence is interpreted. The opposite assumption holds for the succession criteria. They define a precedence function between two segments (a sequence of ordered sentences) on different criteria and formulate the criteria integration task as a binary classification problem and employ a Support Vector Machine (SVM) as the classifier. After the relations between two textual segments are learned, they then repeatedly concatenate them into one segment until the overall segment with all sentences is arranged.

Precedence and succession are interesting criteria, but as we use a human written summary, such information is not available. We adopt the topical relatedness criterion and propose another query-based criterion. Similar to their supervised learning framework, we could also use SVM to combine our criteria to learn the sentence order, as will be discussed as our future work in chapter 6.

2.4.2 Data and Evaluation

Bollegala et al evaluate the method by using the third Text Summarization Challenge (TSC-3) corpus, which contains 30 extracts, each consisting of unordered sentences extracted from Japanese newspaper articles relevant to a query. Each extract has around 15 sentences on average. Two human subjects then arrange the extracts and obtain $30 \times 2 = 60$ sets. Although this corpus has queries associated with each topic, the articles are written in Japanese not English, so we can not use it for our task. Furthermore, their algorithm does not consider any query-related features.

Their system performance was evaluated both manually and automatically. Manual evaluation involves two judges rating the summaries using a four point scale rating: Perfect, Acceptable; Poor; Unacceptable. Automatic evaluation employs rank correlation coefficients...
including Spearman's rank correlation and Kendall's $\tau$ rank correlation. Bollegala et al. also propose a new metric: *average continuity*, which is equivalent to measuring the precision of continuous sentences in an ordering against the reference ordering. It is defined as

\[ AC = \exp \left( \frac{1}{(k-1)} \sum_{n=2}^{k} \log(P_n + \alpha) \right) \]

where

\[ P_n = \frac{m}{N - n + 1} \]

where $N$ is the number of sentences in the reference orderings; $n$ is the length of continuous sentences on which we are evaluating; $m$ is the number of continuous sentences that appear in both the evaluation and reference orderings. $k$ and $\alpha$ are control parameters. Average Continuity becomes 0 when evaluation and reference orderings share no continuous sentences and 1 when the two orderings are identical.

*Average continuity* shares the same concepts as Kendall’s $\tau$, so we will only choose Kendall’s $\tau$ as our evaluation metric. Since manual grading of the system output requires a large amount of human time and effort, we are not able to reproduce this approach. They also use the one-way analysis of variance (ANOVA) to verify the effects of different algorithms and performed Tukey Honest Significant Differences (HSD) test to compare differences among the algorithms.

### 2.5 Barzilay and Lapata

Barzilay and Lapata focus on the evaluation of sentence order quality rather than generating a sentence order directly. We include their work because the evaluation and ordering algorithms are closely related to each other, and the evaluation is also an important part for our thesis work.

This work can be easily extended to the task of sentence ordering by enumerating all the permutations of sentence orderings and finding the best order sequence based on the entity model.

#### 2.5.1 Approach

Inspired by Centering Theory [17], Barzilay and Lapata [5] introduce an entity-based representation of discourse and treat coherence assessment as a ranking problem based on
different discourse representations. A discourse entity is a class of co-referent noun phrases. They use a grid to represent a set of entity transition sequences that reflect distributional, syntactic, and referential information about discourse entities. A fundamental assumption for this method is that the coherence on the level of local entity transitions is essential for generating globally coherent texts. They then take as input a set of alternative renderings of the same article and rank them based on the local coherence. The ranking problem is solved using the search techniques on a Support Vector Machine constraint optimization problem.

Their algorithm outperforms another coherence model based on Latent Semantic Analysis (LSA). They also conduct an experiment to show the contribution of various linguistic features, like syntax, coreference and salience on the model’s performance. They judge the linguistic importance based on syntactic features, but not on semantic features. We use WordNet [14] to capture semantic similarity and local coherence in our work.

2.5.2 Data and Evaluation

The data Barzilay and Lapata use for the evaluation task is the DUC 2003 multi-document summaries produced by human writers and by automatic summarization systems. The training materials contain 96 pair wise rankings with an average summary length of 4.8 sentences.

Coherence ratings were obtained during an elicitation study by 177 native speaker volunteers using a seven point scale rating. The ranking accuracy was measured as the fraction of correct pair wise rankings in the test set.

Since summary contents from different systems are different, this could introduce some bias to the judgment of coherence and the Kendall’s $\tau$ evaluation method can not be used here. But this method might lead to the final automatic evaluation of coherence in the DUC task considering summaries produced by different systems are different. For our thesis task, the content of a summary is predetermined, so we will just follow the Kendall’s $\tau$ approach as discussed earlier.

$^5$They describe two evaluation tasks in the paper. One of them is based on synthetic data which we did not discuss here.


CHAPTER 2. LITERATURE REVIEW

2.6 Summary

In this chapter, we provided an overview of several approaches for sentence ordering in multi-document summarization tasks. A summary chart is shown in Table 2.6. In this section, we are going to give a short summary of the reviewed works according to the ordering criteria, data and evaluation method.

2.6.1 Ordering Criteria

The ordering criteria that have been used so far are chronological information [4, 49, 9], topical relatedness [4, 6, 49, 9], majority sentences position in the original article [4, 6], precedence [49, 9] and succession [9].

Chronological information could help produce good ordering when the information is event-based, and therefore, is temporally sequenced. With query-based summarization, most often the questions concern states or properties, so the chronological approach falls short. Another reason that we do not use a chronology feature is that the data we use consist of human summaries which are lacking chronological information. The advantage of not using chronological information is that the method can also be applied to more generic domains, not relying on the chronological clues provided by news articles.

The majority sentence position feature follows the assumption that the summary structure also follows the structure of the original document set, since multi-document summary captures the main contents among the document clusters. As for query-based summaries, the extracted sentences do not necessarily represent the central meaning of the articles.

The precedence and succession features are interesting criteria, but our input data for sentence ordering are human written summaries, so such information is not available.

Nearly all the work we reviewed has used topical relatedness either implicitly or explicitly. It was first discussed in the corpus study by Barzilay in 2002 [4]. Then in Lapata’s work, the entity-based coherence feature captures the topical relatedness of the lexical level [30]. Additionally, the joint work of Barzilay and Lee has states of HMMs that are topical related sentence clusters [6]. Finally, in the work of Bollegala et al, they consider topical relatedness as an important ordering criteria [9]. So topical relatedness is a good way to capture the cohesion and it will be treated as an important criterion for our sentence ordering task as well.

Barzilay and Lee learned content structure for sentence ordering based on the assumption
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Sentence order do impact the user comprehension</td>
<td>Local coherence can be captured through the probability of lexical and syntactic features of sentence based on the previous sentence. Learn text structure for a specific domain</td>
<td>Word distributional patterns characterize various types of discourse (Content Structure) which can be captured using HMM</td>
<td>Use the machine learning framework to incorporate the four ordering criteria to capture the contingency between two sentences</td>
<td></td>
</tr>
<tr>
<td>2. Multiple acceptable orderings for one document</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Topical related sentences share adjacency relation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rank/Search</td>
<td>Search through weighted precedence graph</td>
<td>Simple greedy search through weighted graph</td>
<td>Ranking by HMM</td>
<td>Agglomerative hierarchical clustering with the ordering information retained</td>
</tr>
<tr>
<td>Features</td>
<td>Majority ordering; Chronological ordering; Topical relatedness augmented chronological ordering</td>
<td>Verbs; Nouns; Structure dependencies</td>
<td>States: Topic clustering (32-95) Transitional Pr: sentence position in the original article</td>
<td>Chronological sequence, Topical relatedness, Precedence and Succession</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>-------------</td>
<td>-----------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Corpus</td>
<td>25 sets of topics, each has 2-3 news articles reporting the same event</td>
<td>BLLIP corpus (30M words) + Barzilay 2002 corpus</td>
<td>5 domains (Earthquake, finance, etc.) Each domain has 100 training/100 testing/20 development set</td>
<td>TSC-3 corpus (Japanese), containing 30 sets of human ordered extracts for multiple document summarization relevant to questions</td>
</tr>
<tr>
<td>Input</td>
<td>Manually selected sentences as extract</td>
<td>Human written articles</td>
<td>Human written articles</td>
<td>Automatic extracted sentences for summary</td>
</tr>
<tr>
<td>Length</td>
<td>8.8 sentences</td>
<td>15.3 sentences</td>
<td>12 sentences</td>
<td>15 sentences</td>
</tr>
<tr>
<td>Eval</td>
<td>Human</td>
<td>3-level grading: Poor, Fair, Good</td>
<td>Human produce summary for upper bound of Kendall's Tau</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Automatic</td>
<td>No</td>
<td>Kendall's Tau (distance between model and original article)</td>
<td>OSO prediction rate; Pair-wise comparison</td>
</tr>
<tr>
<td></td>
<td>Comments</td>
<td>1st paper to use the topical relatedness feature</td>
<td>1st paper to apply Kendall's Tau in automatic evaluation</td>
<td>Learn content structure for ordering</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Apply the machine learning framework to incorporate several ordering constraints</td>
</tr>
</tbody>
</table>
CHAPTER 2. LITERATURE REVIEW

that formulaic text structure facilitates readers’ comprehension [2]. We follow the same assumption and instead of content patterns, we propose a method using questions order to capture the overall text structure, which will be discussed in detail in Chapter 4.

2.6.2 Data

The data for the ordering tasks came from either human written articles [30, 6], or the extracted sentences [4, 49, 9] for a summarization task. Barzilay had humans select the extracted sentences to guarantee the data quality, but such data construction is very expensive. Okazaki et al. use the automatic extracted sentences as the input to the ordering, but their data is in Japanese and they do not give any comments on extracted sentence quality. For our task, we use the human written summaries provided by NIST in DUC 2005. We think it not only captures the summary behavior compared with normal text articles but also guarantees the summary quality. A more detailed description of data will be given in Chapter 4.

2.6.3 Evaluation

The evaluation methods can be divided into two categories: manual and automatic. Although some of the reviewed works [4, 49, 9] involve manual evaluation, investigating information ordering extensively by employing human informants in psycholinguistic experiments is often unfeasible [6]. Recent empirical work [30, 49, 6, 9] on information ordering evaluation adopts automatic methods. In our thesis, we follow the same assumption that a text production method can be evaluated, albeit approximately, by automatically comparing its output with human-defined solutions attested in a corpus of texts, each of which is viewed as a “gold standard” [26]. We evaluate our ordering performance on golden human summaries for DUC 2005 using the most commonly used evaluation measure, Kendall’s $\tau$. Details concerning evaluation will be given in Chapter 6.
Chapter 3

Background – SQuASH Overview

In this chapter we provide an overview of SQuASH system to show how a sentence ordering module fits into a text summarization system, and to introduce some related evaluation issues. The motivation for the study of sentence ordering to answer multiple queries was derived from the development of the SQuASH system, during the Document Understanding Conference (DUC) competition organized by the National Institute of Standards and Technology (NIST). Aside from providing details about SQuASH and its sentence ordering component, we will examine ordering related issues in the context of DUC. The two focuses will be on the input to the ordering module and the evaluation of the output sentence order.

3.1 SQuASH Architecture

SQuASH \cite{43} is a multi-document summarization system, which has been trained and tested on news articles, in response to multiple questions. It was developed by members of the Natural Language Lab from the SFU School of Computing Science in order to participate in the Document Understanding Conference (DUC) summarization competition tasks. The first version of SQuASH system came out in 2005 \cite{43} with an updated system being used in DUC2006 \cite{42}.

The task in DUC \footnote{http://www-nlpir.nist.gov/projects/duc/guidelines.html} is query-based multi-document summarization, which requires synthesizing from a set of 25 news articles a brief, well-organized, fluent answer to a need for information that cannot be met by just stating a name, date, quantity, etc. This task was
Figure 3.1: The Overview of SQuASH Architecture [43].
designed to model real-world complex question answering problems.

The SQuASH [42, 43] system adopts an extractive approach and has three main modules: the Annotator, the Extractor and the Editor. The system architecture is shown in Figure 3.1. The system starts off by annotating the documents and the question text with the Annotator. These annotations are then fed to the next two summarization modules, the Extractor and the Editor. The Extractor focuses on selecting the salient content information to answer the questions. The Editor, which is the last module in the pipeline structure of SQuASH, focuses on linguistic readability by doing sentence compression and sentence ordering. The separation of sentence selection and content structuring might come at a cost: if there is no good ordering given the existing sentences, the system could not go back to the content extraction stage to extract new sentences. But since the content and structure are two important different dimensions of the summary and they are determined by different criteria, we decided to keep these two modules separated. This also makes it possible to plug the content structuring module into other sentence extraction-based summarization systems. Here, we will give a more detailed description of each module, but for a complete system description, please refer to our papers in DUC [42, 43].

3.1.1 Annotator

The Annotator module provides both syntactic and semantic annotations of questions and documents. The syntactic annotations include the output of a statistical parser\(^2\), a named-entity finder and a co-reference resolver [1]. The semantic annotations consist of the output of a semantic role labeler and conceptual information retrieved from WordNet [14]. We use ASSERT [51]\(^3\) to extract the shallow semantic relations, as defined in PropBank, from the syntactic constituents of full parse trees produced by a statistical parser. The precision of ASSERT is 84% and its F-score is 79.4% on the PropBank Corpus. The conceptual information provides ontological relations among words/phrases, thus linking questions and answers at the semantic level. This provides the sentence ordering component the foundation on which to calculate the semantic relatedness between sentences and questions. We will discuss the details of how concept information is identified and used in Chapter 4.

We illustrate the input and output of the Annotator using an example from the first

\(^2\)http://www.cs.brown.edu/~ec/
\(^3\)http://oak.colorado.edu/assert/
**CHAPTER 3. BACKGROUND – SQUASH OVERVIEW**

<table>
<thead>
<tr>
<th>CID</th>
<th>SID</th>
<th>Word</th>
<th>Base Form</th>
<th>POS</th>
<th>Parse</th>
<th>NE</th>
<th>NEID</th>
<th>WS</th>
<th>CID</th>
<th>Pred</th>
<th>SRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>FORMER</td>
<td>FORMER</td>
<td>JJ</td>
<td>(S1(S(NP*</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1087846</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Panamanian</td>
<td>Panamanian</td>
<td>JJ</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>3050149</td>
<td>-</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>leader</td>
<td>leader</td>
<td>NN</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>9489567</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>General</td>
<td>General</td>
<td>NNP</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>9973136</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Manuel</td>
<td>Manuel</td>
<td>NNP</td>
<td>(PERSON</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Antonio</td>
<td>Antonio</td>
<td>NNP</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Noriega</td>
<td>Noriega</td>
<td>NNP</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>'s</td>
<td>'s</td>
<td>VBZ</td>
<td>(VP*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>defence</td>
<td>defence</td>
<td>NN</td>
<td>(NP(NP*</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>17286142</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>against</td>
<td>against</td>
<td>IN</td>
<td>(PP*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>drug</td>
<td>drug</td>
<td>NN</td>
<td>(NP*</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1099321</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>trafficking</td>
<td>trafficking</td>
<td>NN</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>charges</td>
<td>charge</td>
<td>NNS</td>
<td>*))</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>13133691</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>in</td>
<td>in</td>
<td>IN</td>
<td>(SBAR*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Miami</td>
<td>Miami</td>
<td>NNP</td>
<td>(S(NP*</td>
<td>LOCATION</td>
<td>1</td>
<td>2</td>
<td>8943485</td>
<td>(AI*)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>gained</td>
<td>gained</td>
<td>VBD</td>
<td>(VP*</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>1100173</td>
<td>gain</td>
<td>(V*)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>ground</td>
<td>ground</td>
<td>NN</td>
<td>(NP*</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>)</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>this</td>
<td>this</td>
<td>DT</td>
<td>(NP*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>week</td>
<td>week</td>
<td>NN</td>
<td><em>(</em>)</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>14970832</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td><em>(</em>)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>*</td>
</tr>
</tbody>
</table>

Figure 3.2: An example of output from the Annotator.
CHAPTER 3. BACKGROUND – SQUASH OVERVIEW

sentence of the first article in the document set.

Input:

Former Panamanian leader General Manuel Antonio Noriega’s defence against drug trafficking charges in Miami gained ground this week.

Output:

A column format of the sentences output is shown in Figure 3.2. The first column, DID, is the document ID; Second column, SID, is the sentence ID in each document, starting from 0 to n sentence. Column Word is the word in the article with Column Base_Form giving the base form of that word, for example, the base form of said is say. Column POS is the part of speech of the word4. Column Parse shows the parsing information of the sentence with Column NE identifying the Named Entities of the sentence. Column NEID shows Named Entity ID (Same ID refer to the same named entity). Column WS gives the word sense from WordNet and Column CID is the word concept ID. If the two words have the same concept ID, then they are synonyms. Column Pred identifies the predicate (Verb) for Semantic Role Labelling and Column SRL contains the Semantic Role Label tags 5.

3.1.2 Extractor

In the Extractor module, a semantic text graph [35] is constructed based on semantic and syntactic information output from the Annotator to represent the semantic relations for all the documents through shared links and common nodes. Content and sentence selection are performed on the semantic graph.

In the semantic graph, the nodes are concepts, which are identified based on whether a token sequence is a WordNet concept, a named entity, or whether it was a noun. The weights of the nodes are assigned based on the following factors.

- If a concept is present in both a document and a question, then the weight will be increased.
- If a concept in a document has a WordNet type-of relation to a concept in the question, then the weight will be increased.

4For the explanation of each tags in Part Of Speech tagging and parsing, please see http://bulba.sdsu.edu/jeanette/thesis/PennTags.html
5For a detailed description of SRL, please refer to http://www.lsi.upc.edu/~srlconll/home.html
If a concept is a named entity, the weight will be decreased.

If a concept appears in more documents, it will be assigned a higher weight.

The edges of the graph represent the different structural, syntactic, and semantic associations between entities. They include, for example, what semantic arguments, propositions, sentences and documents each concept is found in.

Once the semantic graph is constructed, the SRL propositions within each sentence are ranked based on the concepts they contain and where they are located in the concept graph. This step identifies propositions that make concise use of concepts that are relevant to the summary. Propositions are then iteratively selected based on their ranking. Each selection however also reduces the score of (penalizes) propositions that were similar to the one selected. Once the propositions are selected, their corresponding sentences are selected. In the end the output of the Extractor is the ranked list of selected sentences. The weight and other parameters of the Extractor are optimized on DUC data using the ROUGE score \(^6\), resulting in a good set of candidate sentences for the Editor.

3.1.3 Editor

The task of the Editor module is to produce a summary with high linguistic quality. Specifically, the focus is on sentence compression and sentence ordering. The sentence compression component edits out hypothesized irrelevant content by heuristically dropping certain constituents of the sentence based on the semantic role label and manually designed rules. In this step, any dangling references and context cue information in candidate sentences are removed, which gives a cleaner input to the ordering component. The sentence ordering component selects and clusters sentences based on the questions and lexical cohesion between sentences and attempts to find the most plausible sentence ordering in the summary text.

Component 1. Sentence Compression

The sentence compression component serves as a preprocessing step to sentence ordering. Sentences from multiple documents often contain information that is not only irrelevant to

\(^6\)ROUGE is an evaluation metric that measures the content importance of the summary. It will be discussed in section 3.2.1 in more detail.
the answer of questions but is also specific to the context of the original document. Such contextual information will create great difficulties when it comes to ordering, since the context of a sentence from a specific document is normally not available in the summary. Some of the existing sentence compression algorithms try to rewrite the sentences in a simple way. However, the algorithm normally require a training corpus [28, 54] and since none of the corpora are question oriented, they might remove important constituents that contribute to the answer of the question. To preserve grammaticality with a minimum loss of content information, we only did compression on the surface sentence level and removed the context specific information in the original document.

A series of heuristic constraints are designed for compression as summarized below:

- Remove the temporal \(^7\) and discourse\(^8\) constituents labelled by SRL as ARG-TMP (Temporal markers) and ARG-DIS (Discourse Markers).
- Remove the sub-header from the first sentence of the document, for example PINE RIDGE, S.D. (AP).
- Remove person titles: Mr., Miss, Mrs., etc.
- Remove sentence chunks (\(\leq 5\) words) if it contains original and inflective forms of the verbs such as say, report, argue, suggest, etc. Based on our corpus study, such context specific phrases appear in the news articles with a very high frequency and the content after these phrases often carries important information.
- Clean up the words inside parentheses, dash lines, etc and other formatting symbols in the sentence.
- Remove sentences starting with pronouns, excluding “it”, as pronouns can only be resolved in the original context.

After sentence compression, most candidate sentences stand well in isolation and are suitable for the next ordering step.

\(^7\)Chronological phrases such as yesterday, in this month, next week, Monday, etc. depend on when the news article was published.

\(^8\)Examples of the leading adverbial phrases and certain transitional phrases are Interestingly, Firstly, But, However, And, Yet, Moreover, etc.
CHAPTER 3. BACKGROUND – SQUASH OVERVIEW

Component II. Sentence Ordering

The sentence ordering component first picks the right number of sentences, which are relevant to the supplied questions and satisfy the 250 word length. Then it orders this fixed set of selected sentences. We will discuss these two phases in the following sections.

Phase I: Sentence selection and clustering based on questions

Given multiple questions to answer in a summarization task, the main goal in this step is to select and group every sentence by its contribution to the answer of each question, as we believe that different questions represent different semantic aspects of the topic. The Extractor treats all the questions as a bag of WordNet concepts instead of as individual questions; it is possible that the extracted sentences answering a specific question are always ranked higher than sentences answering other questions. So if we just pick the top \( n \) extracted sentences, it is possible to miss the answer to some questions. To avoid such a situation, the Extractor provides the Editor with sentences amounting to twice the size of the final summary. The Editor then categorizes all the extracted sentences based on a similarity function calculated between the sentences and each question. The distance function is defined as in 3.1:

\[
Sim(S_i, S_j) = \frac{2|\text{concept}(S_i) \cap \text{concept}(S_j)|}{|\text{concept}(S_i)| + |\text{concept}(S_j)|}
\]  

(3.1)

where \( \text{concept}(S_i) \) is a set of concepts in sentence \( i \) and \( |\text{concept}(S_i)| \) is the number of concepts in sentence \( S_i \). \( \cap \) is defined by a set of ontological relations in WordNet, which will be discussed in detail in Chapter 5.

Once the similarity between each sentence and question is calculated, sentences are then grouped into each question cluster according to the similarity measure. The Editor then estimates the number of sentences \( n_i \) to answer each question \( q_i \) within the 250 word length limit based on the proportion between the question clusters. To guarantee a minimum loss of important content, we decided to select the top \( n_i \) sentences for each question \( q_i \) based on their content importance score assigned by the Extractor. In this way, every question is answered while preserving the important content information. The sentence clusters are then ordered according to their corresponding question order.

Phase II: Sentence ordering within the question cluster

After we obtain the question clusters, we order sentences within each cluster based on the contextual information from the original document and the lexical cohesion between
sentences. The algorithm performs the following steps:

- Group sentences from the same original article together, and preserve their original presentation order in the article. The intuition for this is that sentences from the same article often talk about the same events and they are more semantically related. We also observe that many sentences extracted by SQuASH are neighboring sentences in the original article, so we treat such a sentence group as one sentence in the future processing. The maximum number of sentences in one sentence group is 3.

- Pick the first sentence of the summary according to the position of the sentence in the original article. A sentence that is the first sentence in an original article is chosen. If there is more than one sentence that appears as a first sentence in an original article, then choose the one that has higher content importance score, which is assigned by the Extractor. If there is no extracted sentence that is the first sentence in some original article, then pick the sentence that appears in the beginning part of an article. This follows the assumption that the opening sentence in the original article has a higher chance to be a good opening sentence in the summary than sentences that appear in the end of an original article.

- Greedily choose the next sentence group based on its topical relatedness to the current sentence within the same question cluster. We approximated the topical relatedness between sentences based on the similarity function as defined in 3.1.

3.2 Evaluation

Two kinds of evaluations are performed on DUC systems. An automatic evaluation method Recall-Oriented Understudy for Gisting Evaluation (ROUGE)\(^\text{10}\) is used to judge the content salience. The linguistic well-formedness of each submitted summary is manually evaluated by NIST based on five measures: grammaticality, non-redundancy, referential clarity, focus and structure and coherence. Content responsiveness and overall responsiveness of summaries are evaluated manually as well. Each of these manual evaluations is based on a five-point scale: Very poor, Poor, Barely Acceptable, Good and Very good. From the evaluation view of the system, the goal of the Extractor module is to optimize ROUGE and

\(^9\)The "original article" is the source article in which the extracted sentences originally appear.

\(^{10}\)http://www.isi.edu/~cyl/ROUGE/
content responsiveness, and the goal of the Editor is to improve the readability and the linguistic quality of the summary. In the following sections, we will give more detail of the evaluation metrics and how they affect the component design and testing.

3.2.1 Evaluation of the Extractor

ROUGE \[32\] includes measures to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans. The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between the computer-generated summary to be evaluated and the ideal summaries created by humans. Thus ROUGE gives an approximate score for the content selection of sentences, which is the task of the Extractor module. The Extractor successfully utilized ROUGE as the objective function to tune system parameters, and thus to optimize sentence selection from the training data. According to the evaluation results from DUC, our ROUGE score is well above the average performance among all systems. The content responsiveness is the manual evaluation of content selection, and we also achieve a very high score in this measure.

3.2.2 Evaluation of the Editor

1. NIST Evaluation

The goal of the Editor is to improve the readability and the linguistic quality of the summary thus to improve the overall responsiveness of summary. Among the five linguistic well-formedness measures\(^{11}\) given by NIST, Non-redundancy and Focus deal with the content selection which is determined by the Extractor; Grammaticality and Referential clarity are related to the sentence compression component. The Text structure and Coherence measure requires that "the summary should be well-structured and well-organized and that 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." The sentence ordering component is designed to address this measure, which will be the focus of our discussion.

In the NIST human evaluation in DUC 2006, the linguistic scores for our system (SQuASH) were very high, specifically we rank 7th out of 35 systems in structure and coherence. Compared with the performance of same measure in DUC 2005, which is 22nd out of 31 systems,

\(^{11}\)The five measures are Non-redundancy, Focus, Grammaticality, Referential clarity and Text structure and Coherence.
CHAPTER 3. BACKGROUND – SQUASH OVERVIEW

this is a very big improvement. But the score itself can not fully illustrate the usefulness of our ordering methods, since several other factors also affect the quality of the summary in structure and coherence as discussed in the following paragraphs.

<table>
<thead>
<tr>
<th>Content Responsiveness</th>
<th>Percent</th>
<th>Structure &amp; Coherence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Good (5,4)</td>
</tr>
<tr>
<td>5</td>
<td>2.95%</td>
<td>51%</td>
</tr>
<tr>
<td>4</td>
<td>15.76%</td>
<td>31%</td>
</tr>
<tr>
<td>3</td>
<td>30.41%</td>
<td>16.25%</td>
</tr>
<tr>
<td>2</td>
<td>36.06%</td>
<td>9%</td>
</tr>
<tr>
<td>1</td>
<td>6.47%</td>
<td>4%</td>
</tr>
</tbody>
</table>

Figure 3.3: The correlation between content responsiveness and structure & coherence score for the summary.

From the DUC evaluation results, we find that the structure and coherence score has some correlation with the content responsiveness score, as shown in Figure 3.3. Among all summaries produced by 35 systems for 50 topics, which total 1750 summaries, only 2.95% of the summaries are rated as Very Good (Score 5) in Content Responsiveness. Within those Very Good summaries, 51% of them are rated as Good (Score 5 or 4) in Structure and Coherence, 16% are rated as Poor (Score 1 or 2). But for the 6.47% of the Poor rated summaries in Content Responsiveness, only 4% are judged as Good, and 80% of the summaries are judged as Poor in Structure and Coherence. Most summaries are rated as 2 and 3 in content responsiveness. But we can still observe the trend that the better the summary content is, the more possible that the summary is good in Structure and Coherence. This can be better observed from Figures 3.4, 3.5 and 3.6 which show 5 diagrams based on the Content Responsiveness rating of the summary. The x-axis shows the Structure and Coherence score from 1 to 5, the y-axis shows the total number of summaries that fall in each linguistic rating. SQuASH did pretty well on content responsiveness in DUC 2006, ranked 10th out of 35 systems. This might be one of the contributors to the good linguistic score in Structure and coherence.

We also see that some summaries receive very high ROUGE and Content responsiveness scores for content selection, but their overall responsiveness and linguistic qualities are very
Figure 3.4: The correlation between content responsiveness and structure & coherence score for the summary when content responsive score equals 1 (Content Responsiveness I) and 2 (Content Responsiveness II)
Figure 3.5: The correlation between content responsiveness and structure & coherence score for the summary when content responsive score equals 3 (Content Responsiveness III) and 4 (Content Responsiveness IV)
Figure 3.6: The correlation between content responsiveness and structure & coherence score for the summary when content responsive score equals 5 (Content Responsiveness V)

poor. The cause of this poor performance might include factors like unresolved anaphors, irrelevant information, grammar errors etc. However, even when individual sentences stand well in isolation, if they are poorly ordered in the summary, the summary is generally incoherent. Our ordering algorithm achieved a good score in the Linguistic category, especially in the Structure and coherence score, so this showed that the ordering component definitely improve the summaries’ readability to a certain degree.

II. Manual Evaluation

The NIST evaluation does not demonstrate the effectiveness of a sentence ordering module, since so many factors contribute to the quality of the summary. We did a separate evaluation on our ordering module internally by comparing the output summary with the ordering algorithm to the summary without the ordering algorithm. The summary without ordering is a list of sentences ranked by content importance from the compression module. We then gathered coherence judgments from 5 subjects. Each subject was presented with 10 pairs of summaries with their corresponding topic questions. Each summary pair contained summaries with and without ordering. Subjects were instructed to assign a judgment to each summary by a 2-point scale: “good” and “bad”. Summaries from the same topic could
be graded at the same scale, since multiple orderings for the same set of documents are allowed [4]. The study results

The above experiment gives the statistical data to demonstrate the effectiveness of the ordering component. The main change in the Editor algorithm for DUC 2006 compared with DUC 2005 is that we incorporated the sentence clustering based on questions and we adopted a concept based distance measure. This brings the fundamental question that motivates our thesis: How to quantitively evaluate the effectiveness of these two features in sentence ordering?

From the discussions so far, all the linguistic quality evaluation involves human judgment, which make it very expensive to conduct and the results cannot be easily reproduced. So this method does not fit for the large scale evaluation and component optimization. We need to have an automatic evaluation measure that can judge the summary quantitively, so that we can evaluate not only the ordering component, but also the effectiveness of each feature that was used by the component. The remainder of the thesis will address this issue.

3.3 Summary

In this chapter, we reviewed the SQuASH system and its related evaluation issues. The existing evaluation methods provided by NIST do not support component evaluation and optimization, and this brought up the two fundamental questions: How to evaluate our proposed ordering algorithm quantitively and automatically? How to evaluate the effectiveness of each ordering constraint in the algorithm?

In the next few chapters, we will discuss a constrained sentence ordering task where the current features used for ordering in the SQuASH will be applied and evaluated. We will start off by describing the corpus study of human summaries and then providing details of the ordering algorithm and feature selection in Chapter 4. The issues of the experiment design, the automatic evaluation measures and results will be covered in chapter 5.
This chapter proposes and analyzes a query-based sentence ordering algorithm, which is targeted at the sentence ordering task for multi-document summarization in response to multiple questions. The input to the algorithm is a set of sentences that answer the questions. The algorithm orders the sentences to output a coherent summary. Coherence can be defined as a property of well-written texts that makes them easier to read and understand than a sequence of randomly ordered sentences [31].

Several constraints have been proposed for text coherence in summarization [30, 6, 4, 49, 9]. In this thesis, we propose a query order constraint and a lexical cohesion constraint:

- Query order constraint. Query structure is used to determine the text structure. This is a novel constraint and is proposed specifically for the sentence ordering task in query-based summarization.

- Lexical cohesion constraint. Lexical cohesion has previously been used in sentence ordering task to capture the local coherence. We provide a novel similarity function to calculate the lexical cohesion.

These two constraints are incorporated into a revised sentence ordering algorithm in SQuASH in DUC 2006, which was described in Chapter 3. The presentation within this chapter provides greater detail and analysis of the algorithm, with the focus on the description and analysis of the two ordering constraints based on a corpus study.

The chapter is organized as follows: Section 4.1 describes a corpus study of human summaries and gives the motivation for our ordering constraints. Section 4.2 then introduces
CHAPTER 4. SENTENCE ORDERING ALGORITHM

the lexical cohesion constraint. Section 4.3 discusses the query order constraint for sentence ordering. Section 4.4 gives the overview of the sentence ordering method. Section 4.5 summarizes and concludes the chapter.

4.1 Corpus Study of Human Produced Summaries

We create a corpus of human produced summaries in response to questions to identify patterns of orderings that can improve an ordering algorithm. We believe understanding how humans construct summaries will be helpful to the design of the algorithm. Part of the same corpus is also used for the evaluation of the algorithm, which will be discussed in Chapter 5. We will first analyze and discuss the feasibility of adapting the existing corpora to our sentence ordering task, and then give the details of our data corpus.

4.1.1 Collecting the Data

The existing data corpora for sentence ordering tasks come from either human written news articles[30, 6], or extracted sentence sets for summarization tasks[4, 49, 9].

Extracted sentence sets from multiple articles would be very good candidates for the corpus construction, since they are the real input to the ordering component in a summarization application. But due to limitations of the existing automatic summarization systems, system extracted sentences often contain discourse phrases or unresolved anaphora, which would bring noisy input to the sentence ordering task. As we discussed in section 3.5.3, the quality of content selection in the summary has a big impact on the structure and coherence evaluation of the summary, thus a good quality of extracted sentences is the key to good corpus construction. Barzilay[4] has humans do the sentence selection to guarantee the data quality in her corpus construction. She then asks different humans to produce multiple orderings from the same set of extracted sentences. This is a very good summary corpus for finding commonalities among different acceptable orderings of the same information, but these summaries are generic summaries instead of answers to questions. We need a corpus of summaries that address queries, so her corpus is not applicable to our task. Creating a similar data corpus with queries is very expensive and time-consuming, which is not feasible in this thesis. Okazaki et al[49, 9] construct a data corpus for query-based multi-document summarization and they use the system extracted sentences. But their data is in Japanese and no comment is given on the extracted sentence quality in their paper. We have to look
for other alternatives to construct our own corpus.

One alternative is to use human written news articles, which guarantees good content qualities of sentences. Lapata [30] uses the BILLIP corpus, which contains three years of Wall Street Journal (WSJ) articles. Barzilay and Lee [6] collect news articles from five domains in a North American News Corpus. Their corpora have two problems when applied to our task. First, they contain news articles instead of summaries. News articles are often domain specific and they are formulaic. Lapata and Lee [6] are able to learn the content model to capture each domain’s structure. On the other hand, a summary presents information in a condensed form, and it follows a different style of writing compared with news articles. So a corpus study of news article construction might not demonstrate the human behavior in summary construction. Secondly, query based summaries are designed to answer the questions, so the query will affect the summary construction. None of the existing summary corpora in English are query oriented. But one key issue that we want to investigate in this thesis is how the query affects the sentence ordering in summarization. So we need to construct a corpus that has human written summaries in response to multiple queries.

Another advantage of having human written summaries is that it provides the golden summaries as reference summaries in the automatic evaluation, which will be discussed in detail in Chapter 5.

**Corpus Construction**

To identify patterns of how humans construct summaries, we need multiple orderings for the same information. Barzilay [4] asked different humans to produce multiple orderings for corpus construction, but it is too expensive and time consuming for us. Instead, we collect multiple summaries in response to the same set of questions. These summaries are written by different authors and they are composed of different sentences. But these summaries share the same ideas since they are all written to answer the same questions for the same topic. And there are some commonalities among the summary constructions by different authors in response to multiple questions, which will help direct the design of the ordering algorithm.

We collect the human written reference summaries provided by NIST for DUC 2005

\footnote{The five domains are earthquakes, clashes between armies and rebel groups, drug-related criminal offenses, financial reports and summaries of aviation accidents.}
evaluation. Recall that the task in DUC\(^2\) is query-based multi-document summarization, which requires synthesizing from a set of news articles (at least 25 articles) a brief, well-organized, fluent answer to one or more complex questions that could not be met by just stating a name, date, quantity, etc.

There are altogether 50 topics in DUC 2005, each topic consists of one or more questions, a set of 25-50 news articles, and a user profile\(^3\). The user profile is used mainly in the sentence extraction phase, and does not really affect our ordering task, so we will not discuss the user profile in our thesis. These data sets for summarization are part of the TREC\(^4\) news articles from the Financial Times of London and Los Angeles Times collection. NIST assessors\(^5\) choose their topics of interest. Each of these topics has at least 35 relevant documents. The assessors read the documents for a topic, verify the relevance of each, look for aspects of the topic of particular interest, create a DUC topic reflecting the particular interest, and choose a subset of 25 - 50 documents relevant to the DUC topic. The topic is in the form of a question or a set of related questions. The topic statement would describe the required information in only positive terms and it must meet these four criteria:

- The answer can be found in the document sets.
- The answer is complex, requiring about 250 words.
- The statement of the DUC topic should be clear and specific enough so that a person given the topic statement and the corresponding documents will be able to write a good 250-word answer to the questions.
- At least 25 documents must each contribute some material to the answer.

For the evaluation purpose in DUC, reference summaries, which are the ones for our corpus construction, are created by assessors for each topic following the two step guideline provided by NIST.


\(^3\)A user profile contains general or specific, which indicates the granularity of the desired response for each DUC topic. Specific: Answer to identify specific instances of events, people, places, etc. General: A high-level answer that provides generalizations, not the details.

\(^4\)TREC stands for the Text Retrieval Conference. http://trec.nist.gov/

\(^5\)The assessors are educated, adult US natives
• READ and HIGHLIGHT. Each assessor will read the topic and all the documents provided for the topic. Then they use a highlighting pen to highlight the sentences/excerpts containing information that would help satisfy the request for information expressed in the topic. Information that is repeated in more than one document will be highlighted in all those documents.

• WRITE. Write a 250-word summary of the highlighted text that satisfies the information need expressed in the topic. The summary will also be written at the level of granularity requested for the topic, which we will not consider here for simplicity.

The reference summaries are well-organized English text, using complete sentences. They include all information answering the topic, which means the assessors have to generalize some of the information in order to fit everything in 250 words. However, the assessors are not allowed to use specialized knowledge to draw conclusions to make inferences that are not obvious in the documents. So summaries that answer the same set of questions are composed of very similar content as the original articles, but using a very different wording. A total of four references summaries for each of 30 topics and nine reference summaries for each of 20 topics are created.

Within the 50 topics, some topics have only one question, while other topics focus on answering one question mainly compared with other sub questions. We select eight topics where the summary answers each question in a balanced proportion. There are a total of 42 human written reference summaries from the DUC05 testing data as our data corpus, including four summaries for six topics and nine summaries for two topics. For simplicity, we merge some questions for topics with more than two questions, so that each topic has only two questions. A complete list of topics with questions is included in Appendix A.

4.1.2 Corpus Representativeness

To evaluate the effectiveness of a corpus, McEnery [40] proposes the term representative: a corpus is representative if it manifests a range of solutions available to an author. Crucially, evaluation results produced by averaging over a representative corpus are less likely to be biased in favor of a particular information ordering strategy. Most reviewed works make use of a multiple-authored corpus, i.e. texts which are not all written by the same author. In the text corpus we created, six topics have summaries that were written by four different
authors and two topics have summaries by nine different authors. This satisfies the notion of representativeness.

4.1.3 Data Analysis

An investigation of the collected 42 human written summaries shows that most summaries are constructed basically in two ways:

- 28 summaries are constructed by answering questions one by one. Within each question, sentences are constructed by themes. Theme here refers to a subject or topic for discussion as part of the summary. A summary is normally composed of several different themes as most summaries are analytical answers to How and Why questions other than narrative stories.

- 13 summaries structure the answer to questions by themes and within the same theme; sentences are constructed based on the question sequence.

Only one summary (Topic number is Q400b_h07 in DUC2005) is not constructed according to the question sequence. This summary answers the second question first, and its readability is relatively low.

Although for the same set of questions, different humans write summaries in different wordings and in the above two different ways, sentences that discuss the same topic always share an adjacency relation. This conforms to Barzilay's corpus study[4] result that topical relatedness forms local cohesion, which has been used to help with sentence ordering6. The lexical cohesion constraint is also incorporated into our ordering algorithm and will be discussed in detail in the next section.

In the following, we give an example to further illustrate the above observations during the corpus study.

- num:D366I

- title:Commercial cyanide use dangers

---

6Cohesion is defined as a property of “sticking together” different parts of the text. Studies show that the level of cohesion has a direct impact on reading comprehension[18].
1) Cyanide and cyanide compounds, as hydrogen and sodium cyanide, are used in mining processes, electroplating, manufacturing, and other commercial applications. 2) In the heap-leaching process used to extract gold and silver, ore is first placed in a heap on an impermeable plastic pad and then a weak cyanide solution is sprinkled over the ore. 3) The solution collects at the bottom after percolating through the ore and dissolving most of the ore. 4) A similar process is used to process copper mined from open-air pits. 5) Electroplating of metal parts, such as aircraft parts, involves placing the parts in cyanide and acid baths. 6) Hydrogen cyanide is used in manufacturing chemicals, such as tryptophan and other amino acids. 7) Cyanide is also used to recover silver from X-ray film. 8) Cyanide entering air, water, or the soil from commercial activities could have adverse impacts on animals and plants. 9) Waste in the ponds produced by leaching gold from ore and in the pits produced by leaching copper from ore is a source of potential environmental dangers: drinking their toxic solution could kill wildlife; and leaks and overflows from them and their flooding could contaminate groundwater, kill fish and aquatic life, and endanger fisheries. 10) Concerns about potentially harmful effects of mining waste also could cause tourism to decrease. 11) Commercially used cyanide has caused real damage. 12) Work-related deaths have resulted from inhaling cyanide fumes. 13) More than 6,400 migratory birds and other animals were killed in Nevada by drinking water in the cyanide-laced ponds produced by gold mining operations.

Figure 4.1: Type A Summary

- narr: What commercial applications exist for cyanide? What industries is it used in? How is it used? What are its potential dangers? What real damage has commercially used cyanide caused?

Two types of answers are constructed by humans. What we call a Type A summary (Figure 4.1) arranges the answers to one question after the other, and within the answer to each question, sentences are grouped by themes. It first describes the applications of cyanide in mining (Sentence 2 - 4), electroplating (Sentence 5), manufacturing (Sentence 7) etc. Then, the rest of the summary (Sentence 8 - 13) discusses cyanide's potential harmful effects like contaminating ground water, work-related death, and adverse impacts on animals and plants. A type B summary (Figure 4.2) structures the answer by themes and within each theme, sentences are ordered by answering questions one by one. It starts with cyanide's usage in mining (Sentence 2 - 4) and its danger to the environment (Sentence 5 - 7), then it talks about cyanide in the metal plating industry (Sentence 8 - 9) and its potential disposal danger (Sentence 10 - 11), and in the end, it discusses other applications of cyanide, like in tryptophan production and honey gathering.

Our corpus study results suggest that the query order and lexical cohesion patterns are interesting constraints that can be used in sentence ordering.
1) There are two major industries that use cyanide or cyanide compounds. 2) The first use in the mining industry, known as heap leaching, is a very economical and efficient way to separate small traces of gold from rock. 3) Ore is placed on a plastic pad and a weak cyanide solution is poured over it to pull the gold from the rock. 4) The waste from this process is discharged into self-contained ponds. 5) Environmental problems arise during times of heavy rainfall when ponds overflow and run into natural streams. 6) Environmentalists say the runoff could pollute drinking water and endanger salmon fisheries. 7) In addition, birds and other wildlife are endangered when they see the open blue cyanide ponds and stop to drink from them. 8) Another industrial use of cyanide, in the form of hydrogen cyanide gas, is in the plating industry. 9) There is a reported incident of five workers in Indiana dying of asphyxiation when working in an enclosed space in the presence of the cyanide gas. 10) A plating company in Hollywood, California was charged with dumping cyanide into the sewer system and with reckless storage of chemicals. 11) Another plating company in Burbank, California was closed by the EPA for reckless storage of chemicals including hydrogen cyanide. 12) The Japanese use hydrogen cyanide to manufacture tryptophan, an amino acid used as a nutritional supplement. 13) An unusual use of cyanide is to assist Cameroon villagers to gather honey from hives in tall trees. 14) Climbers stun the bees with smoking leaves and a cyanide compound.

Figure 4.2: Type B Summary

4.2 Ordering Constraints

4.2.1 Lexical Cohesion

Local coherence captures text organization at the sentence to sentence level. We assume that global coherence could be achieved by satisfying local coherence using lexical cohesion constraints in ordering [37]. Cohesion is brought about by the referring item and the item it refers to. For example, in the sentences "John ate the apple. He likes apples.", the word he in the second sentence refers back to John in the first sentence, and the word apple is the common subject that the two sentences refer to. There are a number of forms of cohesion such as lexical cohesion, reference, substitution, ellipsis, and conjunction, among which, lexical cohesion is of primary interest in our thesis.

Lexical cohesion involves the selection of a lexical item that is in some way related to one occurring previously. It is established through the structure of the lexis or vocabulary. Reiteration is a form of lexical cohesion which involves the repetition of a lexical item. This may involve the simple repetition of the word but also includes the use of synonymy, hyponymy or meronymy. For example in the sentences John bought a Ford. He loves the car. car is the hyponym of Ford. Meronymy which is the part-whole relationship is also an example of lexical cohesion e.g. the relationship between Canada and Vancouver.

Three broad classes of models have been proposed to calculate lexical cohesion, which
employ word-based, distributional and taxonomy-based similarity measures. Taxonomy-based similarity measures induce word similarity relationships from a manually crafted resource such as WordNet, which have been shown to correlate reliably with human similarity judgment [31].

We follow the taxonomy-based approach and use WordNet to identify synonyms, hyponyms and meronyms. We perform the word-sense disambiguation by using the WordNet::SenseRelate::AllWords module from CPAN\(^7\) and assign a concept ID to each word, which is either a noun, verb or adjective. Recall that the concept ID is defined in WordNet and is used to identify ontological relations between two concepts. An example of word concept ID was shown in Figure 3.2. We then define the similarity between two sentences \(S_1\) and \(S_2\) based on their word concept ID relations in 4.1.

\[
Sim(S_i, S_j) = \frac{2 |\text{concept}(S_i) \cap \text{concept}(S_j)|}{|\text{concept}(S_i)| + |\text{concept}(S_j)|}
\]  

where \(\text{concept}(S_i)\) is the set of concepts in sentence \(i\) and \(|\text{concept}(S_i)|\) is the total number of concepts in sentence \(S_i\). The \(\cap\) between two word concepts is defined as 1 if they satisfy one of the following two conditions and 0 otherwise.

- Two words have the same concept ID, which means that they are synonyms.
- Two words are related by hyponym or meronym relations, which can be identified from a look-up table that is constructed from WordNet.

The value of \(Sim(S_1, S_2)\) ranges between 0 to 1. If two sentences are the same, the value is 1. If two sentences share no similarity, the value is 0. Two sentences are more similar if the \(Sim(S_1, S_2)\) value is bigger and thus they have a higher lexical cohesion.

The main drawback of this measure is that it will indicate low coherence for sentence pairs that have no words or concepts in common based on WordNet relations, even though they may be semantically related.

4.2.2 Query Order

The query order constraint reflects the behavior of how humans ask and answer questions. Given a set of questions, users may have a semantic structure of answers in mind, which

\(^7\)http://senserelate.sourceforge.net/
means they will have an idea of what they expect in the answer even if they don't know the exact content of it.

Hovy\cite{hovy} provides an interesting insight about where questions come from. He stated that

\textit{We are information gathering machines. Our information state is always incomplete. There are things we know, and then the fringes are numerous unanswered questions. The text must address the reader's fringe questions. The informative answer to the current question opens up the new question.}

For example, if the first question is \textit{what is the state of X?}, then the answer to this question provides the background to the next question: \textit{What's the impact of X to people's lives?}

Our corpus study described in the previous section also conforms to Hovy's idea. Summaries are all constructed following the natural question order, no matter what questions direct the structure of the whole summary or a sub-theme of the summary. Since 28 out of 42 summaries have questions that direct the whole summary structure, we will incorporate the question order into our algorithm. Sentences that answer the first question will be placed in the beginning of the summary and sentences that answer the second question will be put in the later part of the summary.

\subsection{4.3 Determining an Order}

Given a set of unordered sentences and their associated questions, our algorithm determines an order of those sentences to form a coherent summary by incorporating the above two ordering constraints. The algorithm consists of two steps as described below.

\textbf{Step 1.}

Follow the query order constraint and cluster sentences according to the questions they answer to give a high-level structure of the summary.

We employ a very simple sentence clustering method. Given a set of unordered sentences and their corresponding questions, we calculate the similarity between sentences and each question using the same formula \ref{equation:1} for calculating the sentence similarity. The only difference is that the similarity calculation includes the one-way meronym relationship, which means that the sentence must have a \textit{part-of} relation to the question. For example, given
the question *Which cities in Canada have an international movie festival?*, the meronym relationship between *Canada* and *Vancouver* will identify the sentence *Vancouver welcomed its 9th international movie festival on September 10th*. This follows the intuition that sentences in the answer often present a specific case to answer a more general question. Once we get the similarity score for each sentence and question pair, we compare the score, and assign the sentence the question label that has the highest score.

One problem with this method is that a question is normally short and contains only a limited amount of key words. Some sentences might demonstrate "no similarity" to all the questions based on the similarity score. In this case, we leave such sentences aside, until all the other sentences have been assigned to a question group. Then we expand the query with its already assigned sentences, and calculate the similarity score between the un-clustered sentences and the expanded query. In this way, we assign the rest of sentences into the question groups.

Once we get the clustered sentence groups, we order the question group based on its corresponding question order. The sentence group that answers the first question is ordered before the group that answers the second question.

**Step 2.**

Find a linear ordering within each question group that satisfies the lexical cohesion constraint.

Given a cluster of $N$ sentences, there are $N!$ possible orders. The set of orders can be represented as a complete graph, where the set of vertices $V$ is equal to the set of sentences $S$ and each edge $u \leftrightarrow v$ has a weight defined as the similarity score $Sim(u, v)$ between sentence $u$ and $v$. Cohen et al.[10] show that the problem of finding an optimal ordering through a weighted graph is NP-complete and they propose a simple greedy algorithm that provides an approximate solution which has been modified for the ordering task in [4, 30]. Similar to their method, we adopt a very simple greedy algorithm.

The algorithm starts by identifying the first sentence in the question group. There are two ways to pick the first sentence. One approach is to pick the sentence based on its position in the original article, which is similar to the *Editor* approach in the SQuASH system. The second approach is to pick the sentence that has the highest content significance score, which is assigned by the Extractor from SQuASH system. Since the test data to the algorithm are human summaries, where no original document position is available, we adopt the second
approach in our thesis. But if we apply this algorithm to other ordering tasks, where the sentence position in the original document is available, the first approach is a better choice. This follows the intuition that a good opening sentence in the original document requires no previous context information, so it is more suitable than some sentences in a lower position of the document, which require the context for introduction. The sentence position constraint has also been used and its effectiveness has been demonstrated in previous works\[4, 30].

Once the first sentence node is picked, the greedy algorithm then selects the next sentence node which has the highest weight with the current node. The selected node then chooses the next sentence that has the highest weight with it. The selected nodes are deleted from the graph. Such operations continue until all sentence nodes are picked and the graph is empty.

After this step, we obtain the linear order of answers for each of the questions. As we already have the question group ordered in step 1, a final summary is generated.

### 4.4 Discussion and Summary

In this chapter we first examined a corpus of human produced summaries and suggested two ordering constraints. We then analyzed the ordering constraints in more detail and incorporated them into a two-step algorithm to solve the ordering task. Before we conclude this chapter, we discuss briefly the strength and weakness of the algorithm as well as its complexity and scalability issues.

We propose a novel order constraint that uses query structure to determine the summary structure by clustering the summary sentences into different groups that answer different questions. This is particularly suitable to the sentence ordering task in the context of query-based summarization.

However, the effectiveness of the query order constraint depends heavily on the quality of the query based clustering. A quantitative study of how much the query based clustering quality affect the sentence ordering results will be described in the next chapter. Some query analysis and expansion techniques which have been done extensively in the question answering (QA) community can be applied here to increase the accuracy of the sentence clustering in the future.

Another advantage of this algorithm is that it can be applied to any domain and any text genre. For comparison, the chronological constraint depends on the publishing time
provided by news articles. The precedence and succession constraints need the context information of the retrieved sentences. The majority sentence position constraint requires that the original documents are from the same domain.

The run time complexity of the algorithm is $O(N^2)$, where $N$ is the total number of sentences to be ordered. The average number of sentences in the summary in our experiments is 15, so the algorithm produces the results almost instantly. The preprocessing step, where questions and sentences are annotated syntactically and semantically, takes a very long time. In DUC, this preprocessing step is done by the Annotator component. Once the documents are preprocessed, they can be stored for future use as we did in DUC2005.

One limitation in the current sentence ordering algorithm is that it only considers each ordering constraint separately. In the future, we are planning to integrate these constraints into a machine learning framework, which would allow us to run both ordering constraints concurrently. This might lead to even better performance of our algorithm.

Once the ordering is produced, we carry out extensive experiments to determine the effectiveness of the algorithm and each ordering constraint in chapter 5.
Chapter 5

Evaluation

In this chapter, we evaluate the algorithm and its ordering constraints as introduced in previous chapter for multi-document summarization in response to multiple questions. We adopt an automatic evaluation measure based on Kendall's $\tau$ method and evaluate our algorithm by comparing summaries generated by the algorithm with human written summaries, which we collected for the corpus study as described in Chapter 4. The evaluation includes a comparison between the baseline system, our proposed method and three alternative methods that have been proposed for the testing of the individual ordering constraints. The evaluation results show that our algorithms outperform the baseline result significantly. We also give evidence concerning the effectiveness of each ordering constraint.

The chapter is organized into three sections. Section 5.1 presents the evaluation criteria and methodology used within the study for sentence ordering. Section 5.2 describes the baseline and several alternative algorithms that are proposed for the comparison of different ordering criteria. Section 5.3 reports the evaluation results of our proposed algorithm and its performance comparison with the alternative algorithms.

5.1 Evaluation Criteria and Methodology

The current evaluation methods for sentence ordering can be divided into two categories: manual and automatic. Although some approaches [4, 49, 9] involve manual evaluation, investigating information ordering extensively by employing human informants in psycholinguistic experiments is often unfeasible [6]. The other problem with manual evaluation is that the results are hard to reproduce which makes it difficult to test different features of
the algorithm. Our DUC experience suggests that a reproducible evaluation measure is very important in component testing and design, so we choose to evaluate our algorithm using the automatic methods.

Some recent evaluations\cite{30,49,6,9} on information ordering adopted automatic methods. We follow these approaches, assuming that a text production method can be evaluated, albeit approximately, by automatically comparing its output with human-defined solutions attested in a corpus of texts. Each of these human written summaries is viewed as a “gold standard”\cite{26}. Kendall’s $\tau$ method has been used in most recent works in automatic evaluation of sentence ordering in text summarization and we use this same method to evaluate our algorithm.

Kendall’s $\tau$ is a measure of correlation, so it measures the strength of the relationship between two variables $X$ and $Y$. It is carried out on the ranks of the data. That is, for each variable, the values are put in order and numbered, 1 for the lowest value, 2 for the next lowest and so on. Kendall’s $\tau$ takes values between $-1$ and $+1$, with a positive correlation indicating that the ranks of both variables increase together while a negative correlation indicates that as the rank of one variable increases the other one decreases.

In our case, the variable $X$ is the natural order of sentences in the summary (from 1 to $N$), which serves as the gold standard of the order. The variable $Y$ is the system generated order. An example of the data representing the original sentence order and system generated order is shown in Table 5.1. The sentence was numbered from -1 to $N$ for the ease of implementation.

Here, let us take a look at how to calculate Kendall’s $\tau$. Kendall’s $\tau$ considers all $\frac{1}{2}N(N-1)$ pairs of data points, say, $(x_i, y_i)$ and $(x_{i+1}, y_{i+1})$. In other words it considers only the relative ordering of ranks between data point pairs $(X, Y)$. The data pair will be either higher, the same, or lower than its matched pair by virtue of their relative values. We define the following terms.

- Concordant: if the relative rank between the $X$’s two data points is the same as the relative rank between the $Y$’s, say, $x_1$ less than $x_2$ and $y_1$ less than $y_2$.

- Discordant: if the relative rank between the $X$’s two data points is opposite to the relative rank between the $Y$’s, say, $x_1$ less than $x_2$ and $y_1$ more than $y_2$.

We then calculate Kendall’s $\tau$ using 5.1.
CHAPTER 5. EVALUATION

Table 5.1: The example of the original sentence order in the summary and system generated order

<table>
<thead>
<tr>
<th>OriginalOrder</th>
<th>SystemOrder</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 5.1: The example of the original sentence order in the summary and system generated order

\[
\tau = \frac{Con - Dis}{Con + Dis} \tag{5.1}
\]

where \(Con\) refers to the number of concordant pairs, \(Dis\) the number of discordant pairs. For the above example in Table 5.1, the \(\tau\) value is 0.272727.

A desired property of Kendall's \(\tau\) measure is its ease of interpretation. Its values range from -1 to 1, reflecting inverse to perfect ordering, respectively. Interpretively, one can view a score as proportional to the probability that a given pair of sentences within a summary is ordered as in the original model; a score of 0, for instance, reflects an even probability that any pair will be in correct versus inverse order; 0.5 means that 75% percent of the sentence pairs are correctly ordered with respect to the original model[8]. Kendall's \(\tau\) seems particularly appropriate for our task. First, the metric is sensitive to the fact that some sentences may be always ordered next to each other even though their absolute orders might differ since the value is calculated by pair wise comparison. Second, the metric penalizes inverse rankings, which seems particularly appropriate for testing our query-based sentence ordering strategy given that flipping the answer to the first question with the answer to the second question seriously disrupts coherence.
CHAPTER 5. EVALUATION

5.2 Baseline and Alternative Algorithms

We evaluate the proposed algorithm by using the same data corpus that we constructed for the human summary study. There are 42 human written summaries in the data corpus. 28 summaries are constructed by answering questions one by one, from which we derived our ordering strategy. One good way to test the effectiveness of our ordering algorithm is to recover the order of these 28 human written summaries after their sentences are shuffled. If our ordering algorithm can generate an order that is close to the original summary order, then it is a good algorithm. The closeness is evaluated using Kendall's $\tau$ method. We then test our algorithms on these 28 human written summaries.

5.2.1 The Baseline

We run the Annotator and the Extractor of the SQuASH system over the input data set to produce the baseline result. Sentences output from the Extractor are arranged by their content importance score. We choose this output over the randomized order as the baseline because the Extractor output is the exact input of the ordering algorithm in the Editor. We want to test if our proposed method will improve the summary quality by rearranging the sentence order.

Recall from Chapter 4 that a human produces each summary from a set of 25-30 news articles in response to two associated questions on a topic. The original input to the SQuASH system is the news article set and the associated questions. But to generate the summary by SQuASH that contains only sentences from the human written summary, we prepared an augmented version of the input document set, containing an additional "document" which is actually the human written summary for that set. Then we changed some parameters of the Extractor to boost the sentence score in the human written summary and select only those sentences. The selected sentences are then arranged by their content importance score to output our baseline summary.

5.2.2 Alternative Algorithms for Comparison

The baseline algorithm provides a way to shuffle the sentences from the original sentence order. The output of the baseline algorithm is then treated as the input to our proposed ordering method to generate a reordered summary. In addition to the baseline and the proposed method, we prepared 3 sets of sentence orderings produced by different algorithms.
CHAPTER 5. EVALUATION

for comparison and testing of the individual ordering constraints alone. We describe briefly the proposed and alternative algorithms as below:

- **Query order (automatically) and lexical cohesion based ordering** *(AC+LC)*
  is the ordering arranged by the proposed algorithm in the thesis. It implements both lexical cohesion and query-based constraints.

- **Lexical cohesion based ordering** *(LC)*
  arranges sentences with only the lexical cohesion constraint.

- **Query order (manually) based ordering** *(MC)*
  involves human effort while doing sentence clustering based on the query. Based on the corpus study described in chapter 4, most sentences answer only one question at a time. We read the summary sentences and assign each of them a question label. Then we cluster sentences with the same question label together, which guarantees a good quality of query based sentence clusters. Current automatic query based sentence ordering algorithms are not very good due to the limited amount of key words in the question. We propose this alternative method to test the effectiveness of the query order constraint given the good sentence clusters based on the query.

- **Query order (manually) and lexical cohesion based ordering** *(MC+LC)*
  uses the manual label to do sentence clustering based on the query. Then it applies the lexical cohesion constraint inside each sentence cluster to produce the final order of the summary.

These algorithms are expected to show the performance of each ordering constraint independently and their individual contribution to solving the sentence ordering problem.

### 5.3 Results and Discussions

Table 5.3 reports the results produced by 5 different algorithms with Kendall’s $\tau$ measure. We calculate the $\tau$ scores of each algorithm by comparing each output summary order with the original order of the human written summary. For a better demonstration, a stacked bar graph is generated as shown in Figure 5.1. X-axis is the summary ID and Y-axis is the $\tau$ score for each algorithm. The graph only shows the system that has the positive $\tau$ score,
which means the system performance is better than the random performance. However, $\tau$ measure alone does not reflect the real performance comparison. So we carried out the paired sample t-test using SPSS\(^1\) (Statistical Package for the Social Sciences) to obtain the results shown in Table 5.3. This test provides evaluation of the difference between two algorithms. If the significance value is less than 0.05, then there is a significant difference between them.

![Comparison between different algorithms](image)

**Figure 5.1: Comparison between different algorithms**

For example, the average performance score of LC is higher than Baseline, so it seems the ordering resulting from LC is of higher quality. But this is not true, since the paired sample t-test returns a value of 0.065, which is higher than 0.05. Consequently, we can not determine which of the two performs better.

On the other hand, our proposed algorithm which incorporates both query order and

---

\(^1\)A comprehensive tutorial of SPSS can be found at [http://www.hmdc.harvard.edu/projects/SPSS.Tutorial/spsstut.shtml](http://www.hmdc.harvard.edu/projects/SPSS.Tutorial/spsstut.shtml)
lexical cohesion constraints obtains a $\tau$ score of 0.1097. Although this value is lower than the performance score of $LC$, it actually makes a significant improvement over the baseline algorithm, because its significance score is 0.031, which is lower than 0.05.

A random sentence ordering would produce a $\tau$ value of 0. The baseline algorithm returns a score of 0.0018, which is slightly better than the random performance. As shown in Table 5.3, all our algorithms except $LC$ outperform the baseline result significantly, which illustrates the effectiveness of the sentence reordering component.

The manual directed query constraint together with the lexical cohesion constraint achieves the best result among all the ordering methods. In fact, by employing humans to do the query-based sentence clustering, the sentence ordering performance is always better than using the automatic clustering method. Its $\tau$ value equals 0.4898, which means that almost 75 percent of the sentence pairs are correctly ordered with respect to the original order.

Although human involvement improves the ordering performance, it is not feasible in a real life application. Our proposed algorithm has been incorporated into the SQuASH system and helped achieve 6th place out of 34 competing systems in the DUC evaluation on the Structure and Coherence measure.

Designing a good query based sentence clustering method will be part of our future work as it significantly affects the performance of the query order constraint. We hope to develop new automatic methods in order to narrow the gap with the human performance.

Our $\tau$ value is in general lower than the results produced in Lapata's experiment[30], where the best performance has $\tau$ value of 0.56. But since the data we use are in a very different domain (one contains news articles and the other is composed of human summaries), we could not compare the results between our algorithm and other previous works that employ the Kendall's $\tau$ evaluation measures.
## CHAPTER 5. EVALUATION

### Table 5.2: Kendall's $\tau$ comparison between different algorithms

<table>
<thead>
<tr>
<th>Topics</th>
<th>Baseline</th>
<th>LC</th>
<th>LC+AC</th>
<th>MC+LC</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>q0436j.h02</td>
<td>0.555555556</td>
<td>0.1666667</td>
<td>0.1666667</td>
<td>0.3888889</td>
<td>0.5555556</td>
</tr>
<tr>
<td>q0370i.h03</td>
<td>0.515151515</td>
<td>0.2727273</td>
<td>0</td>
<td>0.6060606</td>
<td>0.6969697</td>
</tr>
<tr>
<td>q0428e.h02</td>
<td>0.309090909</td>
<td>0.2363636</td>
<td>0.4909091</td>
<td>0.4545455</td>
<td>0.4909091</td>
</tr>
<tr>
<td>q0350a.h04</td>
<td>0.309090909</td>
<td>-0.1272727</td>
<td>0.6727273</td>
<td>0.6727273</td>
<td>0.7818182</td>
</tr>
<tr>
<td>q0400b.h06</td>
<td>0.272727273</td>
<td>0.3333333</td>
<td>0.3030303</td>
<td>0.7575758</td>
<td>0.8181818</td>
</tr>
<tr>
<td>q0360j.h03</td>
<td>0.2</td>
<td>0.3166667</td>
<td>0.45</td>
<td>0.6166667</td>
<td>0.6</td>
</tr>
<tr>
<td>q0400b.h08</td>
<td>0.2</td>
<td>-0.1619048</td>
<td>0.1323810</td>
<td>0.5142857</td>
<td>0.5333333</td>
</tr>
<tr>
<td>q0428e.h01</td>
<td>0.161904762</td>
<td>-0.3523810</td>
<td>0.0666667</td>
<td>0.4476190</td>
<td>0.3142857</td>
</tr>
<tr>
<td>q0446j.h03</td>
<td>0.117647059</td>
<td>0.1323529</td>
<td>0.1323529</td>
<td>0.5735294</td>
<td>0.6911765</td>
</tr>
<tr>
<td>q0400b.h01</td>
<td>0.111111111</td>
<td>0.7333333</td>
<td>0.2222222</td>
<td>0.8222222</td>
<td>0.5555556</td>
</tr>
<tr>
<td>q0374a.h04</td>
<td>0.090909091</td>
<td>0.3333333</td>
<td>0.0303030</td>
<td>0.3333333</td>
<td>0.0909091</td>
</tr>
<tr>
<td>q0400b.h05</td>
<td>0.060606061</td>
<td>0</td>
<td>-0.0303030</td>
<td>0.3636364</td>
<td>0.5151515</td>
</tr>
<tr>
<td>q0400b.h04</td>
<td>0.028571429</td>
<td>0.0476190</td>
<td>-0.230952</td>
<td>0.6571429</td>
<td>0.6952381</td>
</tr>
<tr>
<td>q0436j.h01</td>
<td>0</td>
<td>0.2820513</td>
<td>0.2307692</td>
<td>0.3076923</td>
<td>0.0512821</td>
</tr>
<tr>
<td>q0400b.h03</td>
<td>0</td>
<td>0.0333333</td>
<td>0.1166667</td>
<td>0.3333333</td>
<td>0.3333333</td>
</tr>
<tr>
<td>q0400b.h02</td>
<td>-0.071428571</td>
<td>-0.1428571</td>
<td>-0.2142857</td>
<td>0.3571429</td>
<td>0.7857143</td>
</tr>
<tr>
<td>q0370i.h04</td>
<td>-0.085714286</td>
<td>0.0285714</td>
<td>0.1809524</td>
<td>0.2</td>
<td>0.0476190</td>
</tr>
<tr>
<td>q0366i.h08</td>
<td>-0.099415265</td>
<td>0.0526316</td>
<td>0.0058480</td>
<td>0.309415</td>
<td>0.2163743</td>
</tr>
<tr>
<td>q0370i.h02</td>
<td>-0.120879121</td>
<td>0.4505495</td>
<td>0.2087912</td>
<td>0.6043956</td>
<td>0.5824176</td>
</tr>
<tr>
<td>q0366i.h07</td>
<td>-0.121212121</td>
<td>0.4545455</td>
<td>0</td>
<td>0.7878788</td>
<td>0.4848485</td>
</tr>
<tr>
<td>q0400b.h09</td>
<td>-0.123809524</td>
<td>-0.0476190</td>
<td>0.0666667</td>
<td>0.6571429</td>
<td>0.3904762</td>
</tr>
<tr>
<td>q0436j.h04</td>
<td>-0.126315789</td>
<td>0.2526316</td>
<td>0.2526316</td>
<td>0.6421053</td>
<td>0.4421053</td>
</tr>
<tr>
<td>q0350a.h03</td>
<td>-0.254901961</td>
<td>0.084967</td>
<td>0.1895425</td>
<td>0.3071895</td>
<td>0.1895425</td>
</tr>
<tr>
<td>q0428e.h03</td>
<td>-0.276190476</td>
<td>0.3714286</td>
<td>-0.0857143</td>
<td>0.5428571</td>
<td>0.4666667</td>
</tr>
<tr>
<td>q0350a.h02</td>
<td>-0.309090909</td>
<td>-0.0545455</td>
<td>0.1272727</td>
<td>-0.0181818</td>
<td>0.2</td>
</tr>
<tr>
<td>q0350a.h01</td>
<td>-0.352380952</td>
<td>0.0666667</td>
<td>-0.0285714</td>
<td>0.2190476</td>
<td>0.1047619</td>
</tr>
<tr>
<td>q0366i.h01</td>
<td>-0.418181818</td>
<td>0.1272727</td>
<td>-0.0545455</td>
<td>0.5272727</td>
<td>0.5272727</td>
</tr>
<tr>
<td>q0374a.h01</td>
<td>-0.523809524</td>
<td>-0.5238095</td>
<td>-0.1428571</td>
<td>0.4285714</td>
<td>0.4285714</td>
</tr>
</tbody>
</table>

Average 0.0018 0.1202 0.1097 0.4898 0.4568
### Table 5.3: The Paired Sample Test for different algorithms

<table>
<thead>
<tr>
<th>System Pairs</th>
<th>Sig. Score (2-tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline - LC</td>
<td>0.065</td>
</tr>
<tr>
<td>Baseline - LC+AC</td>
<td>0.031</td>
</tr>
<tr>
<td>Baseline - MC+LC</td>
<td>5.55e-10</td>
</tr>
<tr>
<td>Baseline - MC</td>
<td>9.156e-10</td>
</tr>
<tr>
<td>LC - AC+LC</td>
<td>0.856</td>
</tr>
<tr>
<td>LC - MC+LC</td>
<td>3.029e-08</td>
</tr>
<tr>
<td>AC+LC - MC</td>
<td>1.040e-06</td>
</tr>
<tr>
<td>AC+LC - MC+LC</td>
<td>3.593e-08</td>
</tr>
<tr>
<td>MC - MC+LC</td>
<td>0.314</td>
</tr>
</tbody>
</table>

Note: Table 5.3 describes the results of the Paired Sample Test for different algorithms.
Chapter 6

Conclusion

In this thesis, we described a framework for the sentence ordering task in multiple document summarization in response to multiple questions. We proposed and implemented an algorithm that is particularly well suited to the query oriented summarization. This final chapter first summarizes the approach taken by the thesis in section 6.1. Then it reviews the key contributions to the field of sentence ordering for summarization in section 6.2 and speculates on future research directions in section 6.3. Finally, Section 6.4 presents some concluding remarks.

6.1 Thesis Summary

Chapter 2 began the thesis by presenting a survey of sentence ordering algorithms that were used for ordering information in text summarization. Each approach is discussed in terms of the ordering criteria, the data used for experiment and its evaluation metrics. Various ordering constraints like chronological information, topical relatedness, majority sentence position in the original article, precedence and succession have been proposed and shown to be effective by previous approaches. The topical relatedness constraint, which captures local coherence is treated to be most important and used in all previous research. Most approaches investigate these ordering constraints separately or linearly until the most recent work by Bollegala et al.[9] which provides a supervised learning framework to integrate these different ordering criteria.

The evaluation methods for sentence ordering can be categorized into manual and automatic. Although some of the reviewed works involve manual evaluation, investigating
information ordering extensively by employing human informants in psycholinguistic experiments is often unfeasible. Recent empirical works adopt an automatic evaluation measure based on Kendall's $\tau$ method.

Chapter 3 provided an overview of SQuASH system to show how a sentence ordering module fits into a text summarization system, and introduced some related evaluation issues. The existing evaluation methods provided by NIST do not support component evaluation and optimization, and this brought up the two fundamental questions in the thesis: How to evaluate our proposed ordering algorithm quantitatively and automatically? How to evaluate the effectiveness of each ordering constraint in the algorithm?

Chapter 4 proposed and analyzed a query-based sentence ordering algorithm. A corpus of human produced summaries is constructed and studied, which motivates the two ordering constraints: the query order constraint which uses query structure to determine the text structure and the lexical cohesion constraint that adopt a new similarity function.

Chapter 5 discusses the evaluation of the algorithm and its individual ordering constraints. We adopt an automatic evaluation measure based on Kendall's $\tau$ method and evaluate our algorithm by attempting to reproduce the structure of the human written summaries, which we collected for the corpus. The evaluation results show that our algorithms outperform the baseline result significantly.

6.2 Contributions

The main contribution of this thesis is a sentence ordering algorithm for multiple document summarization in response to multiple queries. It is the first ordering algorithm that was proposed specially for query based summarization.

When compared to the previous sentence ordering works in text summarization, our proposed algorithm can be applied to any domain and any text genre. With respect to running time, the complexity of the algorithm is $O(N^2)$, where $N$ is the total number of sentences to be ordered.

Contributions in this thesis in general include:

- We constructed the first data corpus that contains human written summaries for the query based summarization. We studied the corpus and identified patterns of orderings that can be applied to improve the ordering algorithm.
We proposed a novel ordering constraint that uses query structure to determine the summary structure by clustering the summary sentences into different groups that answer different questions. This is particularly suitable to the sentence ordering task in the context of query-based summarization.

We adopted a new similarity function that uses WordNet to incorporate the semantics similarities across sentences to calculate the lexical coherence between two sentences.

We answered the question *How to evaluate our proposed ordering algorithm quantitatively and automatically?* by using the automatic evaluation method (Kendall’s $\tau$).

We answered the question *How to evaluate the effectiveness of each constraint for sentence ordering?* by setting up different experiments to compare different algorithms with human summary and the baseline.

### 6.3 Future Research

In sentence ordering, when given the current sentence, we pick the next sentence based on their similarity score. The current greedy algorithm implements a search procedure with a beam of width one. In the future we plan to experiment with larger widths (e.g., two or three).

When applying manual and automatic clustering methods, the evaluation results demonstrate that the quality of query based sentence clustering affects the performance of the sentence ordering significantly. So designing a good clustering method is one of our main goals in the future. Several possible approaches can be examined. First, we can apply some of the existing clustering methods like k-mean or complete link clustering. Secondly, there might exist better similarity calculation metrics between sentences. Lastly, we can perform the query expansion, which has been used extensively and successfully in the question answering community. This would increase the amount of concept IDs in queries, which will increase the number of matched sentences for each query.

One limitation in the current sentence ordering algorithm is that it only considers each ordering constraint separately. In the future, we are planning to integrate these constraints into a machine learning framework, which would allow us to run both ordering constraints concurrently. This might lead to even better performance of our algorithm.
6.4 Concluding Remarks

The growing access to large amounts of text data opens more opportunities in information processing. Given a list of complex questions and a set of relevant documents, the task of producing an informative and coherent summary of those documents in response to the questions has attracted a great deal of attention recently.

To solve the problem of organizing information to generate a coherent summary in response to multiple queries, this thesis proposes and implements an algorithm that combines constraints from query order and topical relatedness in human produced summaries. To test the effectiveness of the constraints, we construct a new query-based corpus from the human produced summaries for the Document Understanding Conference (DUC) 2006 evaluation. We then conduct experiments, using an automatic evaluation method based on Kendall's $\tau$, to evaluate and compare the effectiveness of our approaches to others. Our results show that both query order and topical relatedness improve the ordering performance when compared to a baseline method, and a combination of these two constraints achieves even better results.
Appendix A

Appendices

A.1 Test Data Topics

The following are the selected topics of multiple queries from DUC 2005 data. The topic number is the same number from DUC 2005 test data. Each topic has two questions. For the convenience of processing, we merge the questions if there are more than two questions. The title gives the key words of the relevant document set.

- **Topic: d350a**
  
  **Title:** Health-related problems related to working with computers

  **Questions:**
  
  Q1: What health-related problems have been identified with working with computers, how prevalent are these problems?

  Q2: What new health and safety measures, computer equipment, and computer furniture are being introduced into the work place to alleviate these problem?

- **Topic: d366i**
  
  **Title:** Commercial cyanide use dangers

  **Questions:**
  
  Q1: What commercial applications exist for cyanide, what industries is it used in and how is it used?
Q2: What are its potential dangers and what real damage has commercially-used cyanide caused?

- **Topic:** d370i
  
  **Title:** Food labeling
  
  **Questions:**
  
  Q1: What rules have been imposed regarding food labeling and by whom and what doesn't need to be on a label?
  
  Q2: What infractions have been discovered, what penalties imposed and what are continuing problems?

- **Topic:** d374a
  
  **Title:** Nobel Prize Winners in the Sciences and Economics
  
  **Questions:**
  
  Q1: Who are the Nobel prize winners in the sciences and in economics and what are their prize-winning achievements?
  
  Q2: What are common factors in their backgrounds?

- **Topic:** d400b
  
  **Title:** Amazon Rainforest Problems
  
  **Questions:**
  
  Q1: What are the environmental problems facing the Amazon rain forest?
  
  Q2: What steps have been taken to solve them?

- **Topic:** d428e
  
  **Title:** Declining Birth Rates
  
  **Questions:**
  
  Q1: Where are birth rates declining and what are the reasons for declining birth rates?
  
  Q2: What are the effects of a declining birth rate?
• Topic: d436j
  
  Title: Reasons for Train Wrecks

  Questions:
  Q1: What causes train wrecks?
  Q2: What can be done to prevent them?

• Topic: d446j

  Title: Saving Tourists and Tourism

  Questions:
  Q1: What impacts have attacks on tourists had on a government’s economy?
  Q2: What steps have authorities taken to decrease these attacks and what tactics have authorities and businesses tried to lessen the impacts?
Bibliography


