TOWARD ABSTRACTIVE MULTI-DOCUMENT SUMMARIZATION USING SUBMODULAR FUNCTION-BASED FRAMEWORK, SENTENCE COMPRESSION AND MERGING

MOIN MAHMUD TANVEE
Bachelor of Science, Islamic University of Technology, 2011

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

MASTER OF SCIENCE

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

TOWARD ABSTRACTIVE MULTI-DOCUMENT SUMMARIZATION USING SUBMODULAR FUNCTION-BASED FRAMEWORK, SENTENCE COMPRESSION AND MERGING

MOIN MAHMUD TANVEE

Date of Defense: November 29, 2016

Dr. Yllias Chali Supervisor	Professor	Ph.D.
Dr. Wendy Osborn Committee Member	Associate Professor	Ph.D.
Dr. John Zhang Committee Member	Associate Professor	Ph.D.
Dr. Howard Cheng Chair, Thesis Examination Committee	Associate Professor	Ph.D.

Dedication

This work is dedicated to my beloved parents and my dearest wife, Rayhana.

Abstract

Automatic multi-document summarization is a process of generating a summary that contains the most important information from multiple documents. In this thesis, we design an automatic multi-document summarization system using different abstraction-based methods and submodularity. Our proposed model considers summarization as a budgeted submodular function maximization problem. The model integrates three important measures of a summary - namely importance, coverage, and non-redundancy, and we design a submodular function for each of them. In addition, we integrate sentence compression and sentence merging. When evaluated on the DUC 2004 data set, our generic summarizer has outperformed the state-of-the-art summarization systems in terms of ROUGE-1 recall and f1-measure. For query-focused summarization, we used the DUC 2007 data set where our system achieves statistically similar results to several well-established methods in terms of the ROUGE-2 measure.

Acknowledgments

All praise be to the Almighty who gave me the opportunity, determination and strength to do my research. In addition, I believe that I am able to finish this work just because of continuous support and encouragement of my beloved parents.

Now, I would like to thank and express my deep and sincere gratitude to my supervisor Dr. Yllias Chali, Professor, Department of Computer Science, the University of Lethbridge for his continuous support, guidance and encouragement. He not only introduced me to the wonderful world of Natural Language processing, but also provided me with necessary resources, ideas, and time. Without his generosity, patience and support, it would not have been possible to complete this work.

I also thank my supervisory committee members, Dr. Wendy Osborn, and Dr. John Zhang for their time and valuable suggestions.

I am also thankful to my dearest wife, Rayhana who was my sole inspiration and was by my side through all the difficult times. I feel truly blessed for that.

Finally, I would like to express my deep gratitude to all the members of our research group for their support and inspirations. Especially, I would like thank to my friends Mahmudun Nabi, Kawsar Jahan, Sina Golestanirad, and Fatemeh Ghiyafeh Davoodi for their friendship and encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

We live in an information age. Due to the growth of the internet, accessing information has become both simpler and more convenient. Nowadays, whenever we look for information, we use different search engines such as Google, Bing, Yahoo, and so forth, which retrieves and displays a list of documents. Unfortunately, to find the exact information, we need to spend a lot of time on reading these retrieved documents, which are highly redundant and most of the contents are not relevant to the actual need. Automatic text summarization is an effective solution for this problem. Over the last fifty years, the problem of automatic text summarization has been investigated from different perspectives and in various domains. According to Mani and Maybury (1999), automatic summarization is "the process of distilling the most important information from the source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks)".

We use different forms of summarization in our everyday life. For example: before reading a newspaper, we usually read the headline of a news story to determine whether it is worth reading or not; before reading a book, we usually look for the back covers to read the abstract and then consider whether to continue or not; before watching a movie we read reviews. These are all forms of summary.

We can categorize automatic text summarization into different classes based on different criteria as described below.

Indicative vs. informative summarization: Indicative summaries point to concise information without obtaining the detailed information. Newspaper headline and search engine results are the examples of this category. On the other hand, informative summaries aim at obtaining detailed information so that a user does not need to read the documents to know the concepts. Sports news is an example of informative summaries.

Single-document vs. Multi-document summarization: Single document summary focuses on finding the core concepts of a document while the multi-document summary opts to obtain a relevant and non-redundant summary from a document set.

Query-focused vs. Generic Summarization: Generic summarization aims to provide the overview of the information in the documents. To produce the summary it concentrates on covering more information while minimizing overlapped information. On the other hand, in query-focused summarization, an input query is given and a summary is generated by extracting all the information related to that query. A query can be a simple or complex question. For example, a simple question can be like "Who is Donald Trump?" or a complex question can be like "Describe steps taken and worldwide reaction prior to the introduction of the Euro on January 1,1999. Include predictions and expectations reported in the press." To answer this sort of question, query-focused summarization aims to produce a summary which not only contains the important information but also the information related to the given topic in the query.

Extractive vs. Abstractive summarization: Summary generation techniques can be classified into two types: extractive or abstractive method. While extractive methods select some linguistic units from the original document set and append them to form the summary, the abstractive methods aim at generating human-like summaries. In doing so, it requires deep language understanding. Researchers often employ different text-to-text generation techniques such as sentence compression and sentence fusion to generate summaries. Though this approach is more complex compared to the extractive approach, it is possible to generate fluent summaries with more information in a concise manner.

In this thesis, we concentrate on both generic and query-focused multi-document summarization, which are useful in many real world applications. For example, Newsblaster[1] is an automatic news clustering system that presents news summaries from different sources on the web. Other examples, such as information retrieval systems, news headline-generating systems and web search applications, are a few of the many application areas of multi-document summarization research.

1.2 Our approaches

In this thesis, we investigate the effect of different abstraction-based methods in summarization. We know most of the summarization approaches are extractive in nature (Bing et al., 2015), where the system just selects the most important sentences to cover the overall concept of the document set. This approach often leads to redundant information in the summary. Abstraction-based approaches can be a solution for this problem because, in this approach, summaries are formed not only from the source sentences, but also by newly generated sentences constructed from different source sentence fragments. Our summarization model employs the two most popular abstraction-based methods - namely "sentence compression" and "sentence fusion" - for generating a concise sentence set. Sentence compression removes insignificant parts from a sentence, and sentence fusion is a natural language generation technique which merges multiple sentences into a single sentence.

We design a submodular function-based framework for document summarization. Submodular function based frameworks (Lin and Bilmes, 2009, 2010, 2011, Morita and Sakai, 2011) are currently very popular in the field of multi-document summarization. It is known that submodularity naturally occurs in document summarization and we have seen a good performance of this approach (Lin and Bilmes, 2010, 2011). However, to the best of our knowledge, no single work investigates the potential use of sentence compression and sentence merging within a submodular function-based framework. In this thesis, we consider

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

the problem of summarization as a submodular function maximization problem with budgeted constraints, and propose a new submodular function-based framework for document summarization. In addition, we integrate sentence compression and merging with the aim of generating new and concise candidate sentences for the summary. We formulated three submodular functions - namely the importance, the coverage and the non-redundancy, to measure the summary quality. While formulating those submodular functions, we combine both term-based and sentence similarity-based measures.

In earlier works, to solve the summarization problem, both greedy and optimal approaches have been adopted. Optimal approaches are known to be effective in generating high-quality summaries, but it is difficult to obtain a good solution when an immediate response is needed. Also, the optimal approach is not efficient when the problem size is large. On the other hand, in a greedy approach, the algorithm chooses the most relevant sentence in each step to generate a summary. Greedy algorithms are known to be efficient and scalable though it does not often produce an optimal solution.

In our work, we employ a greedy algorithm to maximize our submodular objective function. The reason behind using this algorithm is that the algorithm is efficient and scalable and it can achieve an approximation factor of $(1-1/e)$ of the optimal solution.

1.3 Contribution

This thesis contributes to the domain of generic and query-focused multi-document summarization. Although there is some research on modeling summarization as submodular function maximization, their systems are totally extractive (Lin and Bilmes, 2009, 2010, 2011, Morita and Sakai, 2011). Our formulation is different in the following ways:

- We introduce both a new sentence fusion technique and a sentence compression technique in our submodularity-based document summarization model. This is the first attempt to integrate two abstraction-based methods within a submodular function-based framework for the task of summarization, to the best of our knowledge.

- We consider the Latent Dirichlet Allocation (LDA) based semantic similarity measure to calculate the relevance of a sentence in the summary. Also, we use the semantic similarity measure to reduce redundancy in the summary.
- We introduce a submodular function-based summarization model based on atomic concepts and sentence similarity. Most of the past approaches either use a sentence similarity-based approach (Lin and Bilmes, 2009, 2010, 2011) or a concept-based approach (Takamura and Okumura, 2009; Morita et al. 2013), but this is the first attempt to merge both approaches in document summarization.
- We introduce a new, explicit redundancy measure to obtain non-redundant summaries. Most of the previous works only concentrated on a summary's coverage and had implicit control of redundancy, which does not work well when the topics of the documents are biased. We design a new submodular function for measuring the quality of the summary in terms of non-redundancy.

1.4 Thesis Outline

The rest of the chapters of this thesis is organized as follows:

Chapter 2: We will provide an overview of previous work on summarization, from early extractive approaches of summarization to recent approaches of abstractive summarization. Also, we will briefly describe some necessary background on submodularity.

Chapter 3: We will introduce our proposed summarization approach. First, we will describe the whole process for generic multi-document summarization. We will explain the document processing phase and problem formulation. After that, we will describe the different steps involved during our summarization process in detail. We also explain how we solve the problem of summarization and generate a summary. Moreover, different variants of the proposed approach will be shown. At the end of the chapter, we will describe our approach for query-focused summarization.

Chapter 4: We will introduce the dataset, task descriptions, and the evaluation measures for both generic and query-focused summarization. In addition, we will report our obtained results that are from various experiments that evaluate the variants of our proposed approach. Finally, we will show the comparison of the performance of our system and other state-of-the-systems.

Chapter 5: We will conclude our thesis by suggesting some future directions of our research.

Chapter 2

Background

Text summarization is a well-studied problem in natural language processing since the 1950s. The goal of automatic text summarization is to present the source documents in a concise manner so that a user can understand the contents without reading all the source documents. Text summarization approaches are broadly categorized into two major approaches (Mani, 2001): extractive and abstractive. Extractive summarization aims at obtaining the best subset of sentences from documents so that it can cover all the core concepts in the document set while having minimum redundant information. On the other hand, abstractive summarization aims at obtaining a human-like generation of summaries where the summary sentences are not necessarily coming from the source documents. It requires deep understanding of the source documents.

In this chapter, we review different existing works in the field of automatic summarization. Specifically, we describe some popular extractive summarization techniques and their limitations. We also discuss about different existing abstraction-based summarization techniques and their improvements over extractive summarization. In addition, we discuss some necessary background for the proposed approach, such as submodularity.

2.1 Related Works on Automatic Document summarization

2.1.1 Extractive summarization

Extractive summarization is mostly concerned with identifying sentences that cover the key concepts of the source document set. This technique assigns scores to the original source sentences and picks up the top-ranked sentences to form the summary. Though it does not guarantee that a fluent summary is obtained, it is conceptually simple and does not require deep text understanding (Edmundson, 1969, Carbonell and Goldstein, 1998 and Berg-Kirkpatrick, Gillick, and Klein, 2011).

One of the first extractive summarization systems was proposed by Luhn (1958). Using some simple statistical methods, he was able to select the most important sentences to generate literature abstracts. Another groundbreaking earlier work was proposed by Edmundson (1969) where he first proposed to apply machine learning techniques in summarization research. He extended Luhn's work with several features to determine sentence importance. Some of the features include word frequency in the article, cue phrase, location of the sentence in the article, and title. Both of these works laid a solid foundation for the further research in automatic summarization.

Extractive summarization has gained a lot of interest in the recent years. Most of the recent works are principally based on two important objectives - namely maximization of relevance and minimizing the redundancy in the summary. While the relevance property indicates how much important information is included in the summary, redundancy determines the amount of duplicate information. Different greedy and optimal approaches were applied to generate summaries following these objectives. In the following, we review some of the most notable recent approaches of extractive summarization.

Maximum Marginal Relevance (MMR) by Carbonell and Goldstein (1998) is one of the first approaches that captures the relevance and redundancy measures of a summary. They employed a greedy algorithm that selects the most relevant sentences and ensures non-redundancy by avoiding the sentences that are similar with the already selected sentences in the summary. Though MMR is simple and efficient, its major shortcoming is that the final selection is suboptimal, as the sentence, once selected, is not reconsidered in favour of other sentences (Reidhammer et al., 2010).

Mcdonald (2007) extended this MMR framework and proposed an integer linear programming (ILP) based approach with a global objective function. Though ILP is a good technique to obtain an exact solution in manageable time for small problems, solving an arbitrary ILP is NP-hard and has been proven as impractical in text summarization tasks when the document cluster size is large (Lin and Bilmes, 2010).

Another ILP based framework for query-focused summarization was proposed by Chali and Hasan (2012). Their approach targets to extract the most relevant and query related sentences for a summary. They also built an ILP-based sentence compression model which was proven useful to generate a concise summary, but the ILP solving process was computationally expensive for the joint model of sentence extraction and compression.

Filatova and Hatzivassiluolou (2004) proposed an extractive summarization technique based on events. In the approach, they considered an event as a triplet of two named entities, and a verb which connects these two named entities. Several greedy algorithms for generating a summary were adopted in an attempt to cover the maximum number of important events within the length limit of the summary. This is one of the first approaches where the text summarization problem was formulated as a maximum coverage problem.

Another formulation of summarization as a budgeted maximum coverage problem with knapsack constraints (MCKP) was proposed by Takamura and Okumura (2009). For generating a generic summary, they formulated the problem as an ILP and employed the Branch and Bound algorithm to solve it. They mainly focused on two aspects: coverage and relevance of maximum contents of the documents.

Morita and Okumura (2011) proposed a query-focused summarization with the same problem formulation as Takamura and Okumura (2009), but the key contribution of their work is that they introduced a word co-occurrence graph named “Query snowball” to calculate the query relevance score of each word. In the graph, the innermost nodes are the words which exactly match with the query words. Next, they considered the words which co-occur with the query words in the documents. To measure the query relevance, the words which are close to the center are considered as most important. Also, a query-independent importance score is calculated for all the words by considering the number of their occurrences in the document clusters. To compute the sentence salience score, they used bigrams as the basic unit instead of using unigrams, and ranked the sentences by summing the score of their bigrams. Finally, they employed a greedy algorithm (Khuller et al., 1999) to solve the summarization problem. Neither approach (Takamura and Okumura, 2009 and Morita and Okumura, 2011) properly addressed the redundancy, and by maximizing the coverage sometimes leads to inclusion of less important information in the summary.

The graph-based approach is also popular in document summarization research. One of the first graph-based approaches was LexRank by Erkan and Radev (2004). In LexRank, a document is represented as a graph where the nodes represent sentences and the edges represent similarity between the sentences. The key aspect of this method is to compute the similarity among the sentences and select the sentence which is most similar to other sentences in the document collection. Another variation of LexRank algorithm was proposed by Otterbacher, Erkan, and Radev (2005). The major limitation of these approaches is that they do not consider the syntactic and semantic structure of a sentence while calculating the similarity between them. For this reason, sentences like “The police killed the murderer” are considered as similar to “The murderer killed the police”. However, Chali and Joty (2008) addressed this problem by utilizing the syntactic and semantic sentence similarity measures in their model.

Recently, the concept of submodularity has been used in multi-document summarization research (Lin and Bilmes, 2009, 2010, 2011). Though submodularity is widely used in economics, Lin and Bilmes first introduced it in document summarization research and obtained an efficient solution with a performance guarantee. They formulated summarization as a submodular function maximization problem with budgeted constraints. For representing the documents, they used a similar graph-based approach to LexRank. In (Lin and Bilmes, 2010), they crafted a submodular objective function where a graph cut function was used to measure the summary quality. Furthermore, in (Lin and Bilmes, 2011), they used another class of monotone submodular functions for handling both generic and query-focused summarization. Instead of penalizing redundancy in the objective function, they proposed a diversity reward function to minimize redundancy in the summary. In both works, a greedy algorithm was employed which guarantees a near optimal solution. However, in every case, they ignored individual word importance and used only textual-unit (sentence) similarity based submodular functions, which cannot avoid the redundancy in the summary suitably when the topics of the documents are biased.

Dasgupta et al. (2013) proposed another optimization framework for summarization by generalizing the submodular framework of Lin and Bilmes (2011). In their framework, they expressed the summarization problem as a sum of submodular function for coverage and a non-submodular function called dispersion (inter-sentence dissimilarity). Their motivation for designing a non-submodular dispersion function was that the submodular function cannot capture redundancy constraints, which depend on pairwise dissimilarities between the sentences. They applied a greedy algorithm to obtain an approximately optimal summary.

Recent work of Christensen et al. (2013) introduced another measure of an extractive summary: coherence. While most traditional multi-document summarization techniques mainly focus on coverage of the content and less redundancy, they introduced a coherent summarization system. In their approach, an approximate discourse graph is proposed where each node denotes a sentence and each edge between the nodes denotes the discourse relationship between the sentences. The weight of the edges of the graph is determined using a number of features such as deverbal noun reference, event/entity continuation, discourse markers, co-referent mention and so on. Positive weights denote a discourse relationship between the sentences, while no edge between two nodes that implies two sentences are disconnected. While selecting the sentence to form a summary, they emphasized this discourse relationship between the sentences, as well as salience and non-redundancy.

In another work, Christensen et al. (2014) introduced a new approach to handling a large number of documents for summarization, called hierarchical summarization. Unlike the traditional multi-document summarization techniques that take a few documents and make a short summary form the document set. Their summarization system can generate a long summary like Wikipedia from a large number of documents. In their hierarchical summary, the top label of the summary gives the overview of different topics discussed in the document set. A user can obtain more details on any topic by clicking on any topic from the first label. This allows the user to learn about a topic as much as they want. The whole process is done in two steps. Hierarchical clustering clusters all the sentences in the input documents by time. In the next step, sentences are selected that best represent the cluster. To assess the summary quality, they formulated summarization as an optimization problem where they emphasized a summary's salience, coherence, and non-redundancy. Finally, they solve the problem using a beam search.

2.1.2 Abstractive summarization

Most of the research on document summarization focuses on extractive methods where summaries are formed by selecting most relevant sentences. Though this approach is simple, it has a number of limitations. For example, this approach always either selects a sentence in the summary or totally exclude it from consideration. For this reason, sentences with partly relevant information can be selected as a summary sentence. This leads the inclusion of insignificant information which degrades the summary quality. Moreover, by just concatenating some important source sentences, mostly it is not possible to generate a coherent summary. On the other hand, the abstraction-based approach aims at generating summaries by deeply understanding the contents of the document set and rewriting the most relevant information in natural language. Though it is a complex task to generate a fully abstractive summary, researchers most often try to modify the source text sentences to generate sentences which are either a shorter version of the original ones or a combination of information from multiple sentences. Two recent abstractive techniques are most commonly used to accomplish this task: sentence compression (Knight and Marcu, 2000) and sentence fusion (Barzilay and McKeown, 2005).

Sentence Compression:

Sentence compression is a process of shortening sentences by removing or deleting insignificant part of a sentence. Using this, it is possible to generate grammatical sentences while preserving the most important information of the original sentence (Jing, 2000). It has been successfully used as a step of summarization with many extractive strategies (Gillick et al., 2009). In this section, we briefly summarize some of the notable research on sentence compression.

One of the earliest works on sentence compression was proposed by Jing et al. (2000). They proposed a supervised approach where a sentence is considered as a number of phrases. Their target was to identify the insignificant phrases and remove them to generate a sentence which is grammatically correct and contains the core information. First, they determined the phrases which are grammatically correct. In addition, lexical analysis was done to determine the importance of each word. Finally, a sentence compression parallel corpus was used to calculate the probability of deleting a phrase from a sentence.

Knight and Marcu (2002) proposed another compression technique using a noisy channel model, which depends on a parallel corpus to build the compression model. The model is comprised of three models - namely source model, channel model, and decoder model. The source model takes a source sentence q and target compression r and calculates the grammaticality correctness probability $p(r)$ using a language model. The channel model computes $p(q|r)$, which is the probability of how likely the sentence q can be an expansion of the compressed sentence r. Finally, the decoder model looks for a compressed sentence that maximizes $P(r).P(q|r)$.

Another compression model was developed by Clarke and Lapata (2008). They formulated compression as an optimization problem and used integer linear programming (ILP) to infer optimal compression. Three compression models were developed - namely unsupervised, semi-supervised and supervised. The unsupervised model uses a language model to determine insignificant n-grams within a sentence. The semi-supervised model combines a language model with a word significance score. The supervised model uses a large parallel corpus to train a model. Also, some hand-crafted constraints are employed to improve the quality of the compressed sentences.

Filippova and Strube (2008) used an unsupervised sentence compression approach for summarization. They used a dependency tree representation for a sentence and applied compression techniques to prune subtrees to get the final compressed sentence. In their work, compression was considered as an optimization problem. Integer linear programming (ILP) was adopted to find the smallest dependency graph that maximizes the objective function. They also ensured the grammaticality of the output sentence using a tree linearization step. The uniqueness of their method is that they did not use any language model to test grammaticality.

Recently, a joint model of sentence compression and extraction was proposed by Berg-Kirkpatrick et al., (2011). They formulated their joint model as an ILP problem where the objective function is comprised of two factors: coverage and compression. For the compression, they introduced a new annotated dataset of extracted and compressed sentences. They also introduced a number of features to identify the individual sub-trees for deletion in the constituency parse tree for each sentence. Finally, they solved the summarization problem by using both ILP and a fast approximate scheme and achieved similar performance.
In our proposed approach, we follow the features introduced in this work for the compression task. However, our summarization model is totally different from theirs, where we formulated summarization as a submodular function maximization problem and employed greedy algorithms.

Sentence Fusion

Sentence fusion is a text-to-text generation technique that combines sentence fragments from multiple sentences to create a more informative sentence. It can improve the summary quality by covering more information in a concise manner and reducing redundancy. However, generating new sentences using sentence fusion is difficult as it often leads to ungrammatical sentences. In the following, we review some of the most notable sentence fusion-based abstractive summarization approaches.

Barziley and McKeown (2005) first introduced sentence fusion in text summarization research. In their summarization model, they first clustered the related sentences using machine learning techniques. For each sentence in the cluster, a dependency parse tree is generated and new sentences are constructed by fusing those trees. Finally, the best-fused sentences are selected via ranking against a language model.

Filippova (2010) introduced another abstraction-based technique where both sentence compression and fusion were applied to generate new concise sentences for the summary. The key assumption of the work is that the redundancy among similar sentences provides a robust way to generate informative and grammatical sentences. At first, a set of related sentences is clustered to construct a word graph. In the graph, each node denotes a word and an edge between the two nodes denotes the adjacency relation between them. While adding the sentences in the graph, if a word appears more than once in the graph, the last word is mapped with the word already existing in the graph. Edge weight is assigned by counting the frequency of two words occurring consecutively in different sentences. After that, the edge weights are inverted and multi-sentence compression is done by finding the lightest path from the graph. In her first approach, she did not consider individual word importance while ranking the fused sentences. In another approach, Filippova considered individual word importance and used another scoring function to rerank all the paths. The final scores of fused sentences were also normalized over their lengths. Finally, the lightest path is chosen as a summary from the candidate-fused sentences.

Boudin and Morin (2013) improved Filippova's (2010) work by incorporating keyphrase extraction and generated more informative summaries. Like Filippova (2010), they first constructed a word graph. The major limitation of Fillipova's work is that it fails to capture a lot of important information in the summary. To improve the system, they extracted keyphrases using an unsupervised approach based on the work of (Wan and Xiao, 2008). While Filippova (2010) only considered finding the shortest path in the graph as a summary, Boudin and Morin (2013) considered a large number of shortest paths as candidate solutions. Finally, they reranked the paths based on the key phrases they contain.

Ganesan et al. (2010) proposed a graph-based approach to producing abstractive summaries from highly redundant opinions. They used only shallow NLP techniques such as word order and word redundancies as clues to produce informative summaries. At first, a graph is constructed by adding all the sentences in the document set. In the graph, nodes represent the words and for every distinct word, only one node is created. So, if there are multiple sentences containing the same words, the sentences pass through the same node in the edges. By this graph construction, every path in the graph is either an original sentence or a fused sentence. To generate a summary, they ranked all the paths based on the redundancy they contain. The path which has the most redundancy is considered as the best path. Finally, the best paths are selected with minimum duplicate information among them.

Cheung and Penn (2014) proposed another sentence enhancement technique for automatic summarization. First, the system clusters the core input sentences which are very close to each other and constructs a sentence graph with the core sentences. Then the graph is expanded by including the non-core sentences that describes the same event. These non-core sentences are identified by an event coreference module. They formulated summarization as an optimization problem where they considered extracting the tree from the graph that maximizes the objective function. Finally, they applied a tree linearization step to generate grammatical sentences.

Recently, Bing et al. (2015) proposed an abstractive multi-document summarization framework by selecting important phrases and merging them. Unlike most of the approaches, which take sentences as the basic syntactic unit, they considered the noun and verb phrases as the basic unit. First, they cluster a pool of coreferent noun phrases (NP) and verb phrases (VP) from the input documents using a coreference resolution engine. These coreferent NPs and VPs were later used for new sentence generation. For calculating the salience score of the phrase, they adopted a concept-based weighting scheme, which incorporates position information of the phrases. While fusing the phrases to construct a sentence, they followed a simple heuristic. The heuristic is that a new sentence is constructed by at least one noun phrase (NP) and one verb phrase (VP), where NPs and VPs may come from different input sentences. They formulated summarization as an optimization problem where the salience scores of the phrases are maximized. Finally, they employed ILP to solve the optimization problem. The main problem with this approach is its time inefficiency.

This is another closely related work with our proposed approach. But the major difference between this approach and ours is that we considered sentence as the basic linguistic unit where they used phrases. Their approach maximizes the salience score of phrases to generate summaries using ILP where our summarization model is submodular and we employ a greedy algorithm to accomplish the task.

2.2 Submodularity

In mathematics, submodularity is an important property of set functions. It has been used in a wide variety of applications such as game theory, information gathering (Krause and Guestrin, 2007), combinatorial optimization (Edmonds, 1970), image segmentation (Boykov and Jolly, 2001; Jegelka and Bilmes, 2011a) and operational research. Recently, submodularity has also been considered in Natural Language Processing research (Lin and Bilmes, 2009, 2010, 2011). As we have used submodularity in our proposed summarizing model, we provide a background on submodularity in the following section.

Definition:

In mathematics, a set function takes a set as input and outputs a value. Submodularity is an important property of a set function. Suppose, there is a finite ground set of objects $U = \{u_1, u_2, u_3, \ldots., u_n\}$ and let $f : 2^U \rightarrow R$ be a function which assigns a real value for each subset $S \subseteq U$. We can say, the function $f(.)$ is submodular if

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \qquad \forall A, B \subseteq U$$

Function $f(.)$ is called normalized if $f(\emptyset) = 0$ and it is monotone submodular if $f(A) \leq f(B)$ when $A \subseteq B$.

Submodular function exhibits another important property called *diminishing returns*. The diminishing return property states that adding an element in a smaller set has more gain than adding it to any of its supersets. We can say, $f(.)$ is submodular if for any $B \subseteq A \subseteq V$ and $p \in V \setminus A$,

$$f(A \cup \{p\}) - f(A) \leq f(B \cup \{p\}) - f(B)$$

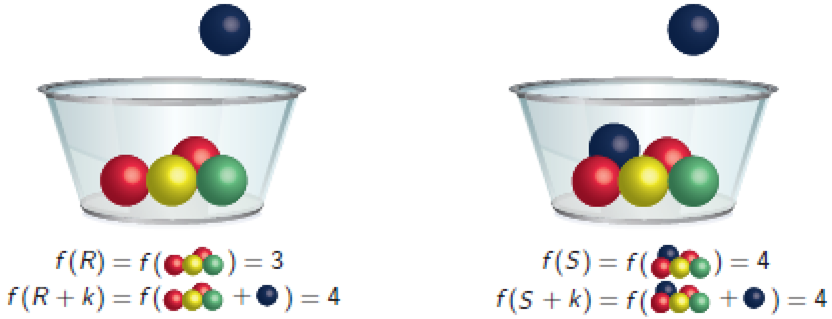

Figure 2.1: Example of submodular functions (Lin, 2012)

We can understand the concept of submodularity and diminishing returns from an example shown in Figure 2.1. In the figure, there are two containers. The left container contains 4 balls with 3 different colors, and the right one contains 5 balls of 4 different colors. Suppose, a function $f(:)$ counts the total number of the distinct colored ball in the container. Also, note that all the balls in the left container are present in the right container. So we can say that the right container is the superset of the left. Now if we add a blue ball in both of the containers, the number of different colored balls becomes 4 for the left container while the right container has the same number of different colored balls. That means the function f has an increment of 1 for the left container where for the right container, the increment is zero. This illustrates that by adding a new element in both sets, the subset will always have at least the same gain of the superset. This is the concept of diminishing returns. So we can say, the function f that counts the distinct colored balls is submodular. This property can be applied to summarization purpose too, because adding a new sentence in a summary that contains sufficient information should have a smaller gain compared to a smaller summary with less information (Lin and Bilmes, 2010, 2011).

2.3 Summary

In this chapter, we have summarized different techniques of summarization. First, we reviewed extractive approaches. We also discussed different abstraction-based approaches such as sentence compression and fusion. Some related works of those approaches were also reviewed. In addition, we briefly explained the concept of submodularity which is related to our research. In the next chapter, we are going to explain our summarization techniques, where we employ a submodular function based framework and different abstraction-based techniques for summarization to overcome the shortcomings mentioned in previous approaches.

Chapter 3

Toward Abstractive Document Summarization

3.1 Introduction

In this chapter, we introduce our proposed summarization system and discuss different steps to perform automatic summarization. While designing our system, we applied both extractive and abstractive approaches and designed a hybrid system taking advantages from both. The main problem of a fully extractive system is that it only picks the most important sentences to form the summary. For this reason, sometimes summary quality can be degraded if longer, but partly relevant sentences exist in the summary, possibly preventing the inclusion of other important sentences (Martins and Smith 2009).

Two abstraction techniques, namely sentence compression and fusion, could remedy this problem. Sentence compression mostly deals with a single sentence and tries to improve it by reducing or removing the less important parts of the sentence. Using this technique, it is possible to make space in the summary for more relevant concepts. On the other hand, sentence fusion merges multiple sentences into a single sentence. In the past, most well-known summarization approaches concentrated on extractive approaches for their simplicity and speed but did not pay much attention to abstractive techniques to improve the summary quality. This is our motivation for proposing a summarization system where we integrate different abstraction-based techniques with an extractive summarization to produce near optimal summaries.

Our system defines the summarization task as a submodular function maximization problem with budgeted constraints. We designed and implemented our proposed summarizer, which utilizes three important factors of a good summary. The factors are importance, coverage, and non-redundancy. For each property we designed a submodular function and then applied an approximate algorithm with a performance guarantee to obtain a near optimal summary. In addition, we use two abstraction factors – sentence compression and merging – to obtain summaries that can cover more content more concisely. We focused on both generic and query-focused summarization and designed two systems for each of them.

3.2 Notations and Definitions

In this section, we introduce the most important notations used in our proposed automatic document summarization model.

Table 3.1: List of notations in automatic document summarization.

notation	description.
D	set of multiple documents containing sentences
U	set of all sentences in the document set
S	summary
S_i	ith linguistic unit (sentence)
w	conceptual unit (noun, verb, and adjectives)
B_{max}	summary length
st	sub-tree
$PT(s)$	parse tree

3.3 Document processing

Pre-processing is an important step for an efficient summary generation. In this section, we give a detailed description of the different pre-processing steps that are done before the summarization.

3.3.1 Sentence splitting and tokenization

Our first pre-processing task is to break down a document into several linguistic units. Here we consider the sentence as the basic linguistic unit of a document. We used Stanford CoreNLP[2] (Manning et al, 2014) to do both sentence splitting and tokenization. While the sentence splitting operation separates sentences (Jurafsky and Martin, 2000), the tokenization operation segments text into tokens. These tokens include all types of words, punctuations, numbers, etc.

3.3.2 Case folding

We use a simple strategy of case folding: we reduce all the letters to lower case, to allow a better estimation of token frequency, which improves the scoring of tokens while doing the sentence ranking for the summary.

3.3.3 Removing stop words

Stop words[3] are basically the set of words most commonly used in every language. These extremely common words usually have little importance in the summary generation, and by removing these words we can focus more on the important and topic-related words. In this project, we used a stop word list of 598 words to obtain only topic-related important words from the documents.

[2] http://stanfordnlp.github.io/CoreNLP

[3] In general, there is no defined list of stop words in English, but we have used a stop word list of 598 words which is shown in Appendix A.

3.3.4 Part of Speech tagging

For the purposes of tagging each word of a sentence with its part of speech, we use the Stanford POS tagger (Toutanova et al., 2003), which uses the most popular Penn Treebank POS tag set[4]. Since we consider only atomic terms describing events (verb), named entity (noun), and adjectives as conceptual units for sentence ranking, the POS tagger has been used to obtain all the nouns, verbs, and adjectives from the documents.

An example of part of speech tagging is illustrated below.

Input Paragraph : Cuban President Fidel Castro said Sunday he disagreed with the arrest in London of former Chilean dictator Augusto Pinochet, calling it a case of "international meddling". Castro had just finished breakfast with King Juan Carlos of Spain in a city hotel. Pinochet, 82, was placed under arrest in London Friday by British police acting on a warrant issued by a Spanish judge.

Output Paragraph after Part of Speech tagging: Cuban/NNP President/NNP Fidel/NNP Castro/NNP said/VBD Sunday/NNP he/PRP disagreed/VBD with/IN the/DT arrest/NN in/IN London/NNP of/IN former/JJ Chilean/JJ dictator/NN Augusto/NNP Pinochet/NNP, calling/VBG it/PRP a/DT case/NN of/IN "international/JJ meddling/NN". Castro/NNP had/VBD just/RB finished/VBN breakfast/NN with/IN King/NNP Juan/NNP Carlos/NNP of/IN Spain/NNP in/IN a/DT city/NN hotel/NN. Pinochet/NNP, 82/CD, was/VBD placed/VBN under/IN arrest/NN in/IN London/NNP Friday/NNP by/IN British/JJ police/NNS acting/VBG on/IN a/DT warrant/NN issued/VBN by/IN a/DT Spanish/JJ judge/NN.

3.3.5 Removing punctuation

We also remove punctuation from the token list as they do not contribute anything to the sentence ranking process.

[4] http://www.comp.leeds.ac.uk/ccalas/tagsets/upenn.html

3.3.6 Stemming

Stemming is a way of reducing inflected words to their root form. It is commonly used in the NLP and document summarization fields. In this project, we use the Porter Stemmer[5] (Porter, 1980) for stemming. The reason for using stemming is that it helps better detect the correlation between words. An example of stemming is shown below.

Input sentence: Cambodian leader Hun Sen on Friday rejected opposition parties' demands for talks outside the country, accusing them of trying to "internationalize" the political crisis.

Stemmed sentence: cambodian leader hun sen on fridai reject opposit parti demand for talk outside the countri accus them of try to internation the polit crisi.

3.3.7 Document coreference resolution

In summarization research, it is really helpful to know which terms refer to which entities. For example, a person can be referred to by their full name, first or last name, or by a pronoun. For coreference resolution purposes, we used the Stanford Coreference Resolution package (Lee et al., 2013) to extract all the coreference groups with their coreference mentions from the text document. Consider the following example sentences.

Cambodian leader Hun Sen rejected opposition parties' demands.

Hun Sen was not home at the time of the attack.

Sen complained that the opposition was trying to make its members' return an international issue.

He has rejected the opposition's reservations.

Here, all the sentences discuss the same person Hun Sen, but not with the same set of words. After coreference resolution, the output is as follows:

[5] http://tartarus.org/martin/PorterStemmer

```
<coreference>
<coreference>
<mention representative=“true”>
<sentence>1</sentence>
<start>1</start>
<end>5</end>
<head>4</head>
<text>Cambodian leader Hun Sen</text>
</mention>
<mention>
<sentence>2</sentence>
<start>1</start>
<end>3</end>
<head>2</head>
<text>Hun Sen</text>
</mention>
<mention>
<sentence>3</sentence>
<start>1</start>
<end>2</end>
<head>1</head>
<text>Sen</text>
</mention>
<mention>
<sentence>4</sentence>
<start>1</start>
<end>2</end>
```

```
<head>1</head>
<text>He</text>
</mention>
</coreference>
<coreference>
<mention representative="true">
<sentence>3</sentence>
<start>4</start>
<end>6</end>
<head>5</head>
<text>the opposition</text>
</mention>
<mention>
<sentence>3</sentence>
<start>10</start>
<end>11</end>
<head>10</head>
<text>its</text>
</mention>
</coreference>
</coreference>
```

Here we can see the coreference resolution engine extracts “Cambodian leader Hun Sen” as the representative of the coreference group. Then, it adds different coreferent mentions such as “Cambodian leader”, “Hun Sen”, “Sen”, and pronouns such as “he” in the same cluster. From each cluster, we replaced the coreference mentions with the coreference representative of the coreference group. Finally, we updated the token stem list.

3.4 Problem Formulation

We divide the whole task of summarization in two phases.

- Document shrinking
- Document summarization

In the document shrinking phase, we employed different abstraction-based techniques, such as sentence compression and sentence merging. The main motivation of applying these techniques is that it allows to remove the insignificant sentence parts from sentences all over the document set. Also, by deploying the sentence merging, it is possible to merge important information from different source sentences. By this way, this phase ensures to provide a concise and informative sentence set. In the document summarization phase, we formulated summarization as a submodular function maximization problem with budgeted constraints. In the past, several approaches have been proposed where an extractive summarizer was improved with sentence compression. However, there is no single work where an both sentence compression and fusion have been employed with a submodularity-based model for the summary generation. In the following, we explain the two phases of our proposed system for generic multi-document summarization.

3.5 Generic Summarization

3.5.1 Document shrinking

In this phase, we used sentence compression and sentence merging to prepare a better and more concise document set for summarization before approaching the actual summarization.

Sentence Compression

Sentence compression is a technique of shortening sentences which can be used with the extractive system to improve summary quality. Using this technique, we remove or delete unnecessary phrases from a sentence and save space to include more important information in the summary (Knight and Marcu, 2002). While extractive summarization systems either include a sentence or completely exclude it in the summary, incorporating sentence compression allows us to include part of the sentence in the summary. Consider the following example sentence as a candidate sentence[6] of the summary:

"*According to a newspaper report*, a total of 4,299 political opponents died or disappeared during Pinochet's term."

In this sentence, the part shown in the italic font is not carrying much significance. So if this part is removed, we obtain a compressed sentence like "a total of 4,299 political opponents died or disappeared during Pinochet's term" which saves some space in the summary. Also, after removing this insignificant part, the sentence is still grammatically correct and it preserves the significant information. So, from this example we can say, sentence compression is really an effective technique in the summarization task when the summary length is fixed. Our system applies Berg-Kirkpatrick's (2011) sentence compression technique to identify and remove the insignificant parts from the sentence. In the following, we describe the sentence compression technique.

Sentence compression based on Berg-Kirkpatrick et al., (2011) technique:

At first, we generated a constituency parse tree for every sentence in the document set. We used the Berkeley parser[7] (Petrov and Klein, 2007) to obtain the constituent parse trees. Figure 3.1 shows a constituency parse tree for the above example sentence. A constituency

[6] The example sentence is taken from DUC 2004, topic d30003t

[7] https://github.com/slavpetrov/berkeleyparser

parse tree is an ordered, rooted tree which represent a sentence according to some context-free grammar. In this tree structure, the interior nodes denote non-terminal categories of the grammar and leaf nodes represent terminal categories (tokens of a sentence).

In the second step, we find the insignificant parts from each of the constituency parse tree and delete them to obtain compress sentences. To do so, we consider that a parse tree is a set containing its subtrees. First, we divided every parse tree into all possible subtrees and obtained a set for each of the parse trees. Suppose, X is the set of subtrees for the parse tree $PT(s_i)$ where $X = \{st_{1i}, st_{2i}, ..., st_{mi}\}$. Here, m is the number of subtrees for the sentence s_i and st_{ji} is the jth subtree of parse tree $PT(s_i)$. Then, we delete one or more subtrees from a parse tree. Therefore, compressing a sentence is nothing more than deleting subtrees from a parse tree. For finding the insignificant parts, we use Berg Kirkpatrick et al.'s (2011)[8] deletion features. Berg-Kirkpatrick et al., (2011) introduced 13 features, which are trained using the TAC data set. Table 3.1 shows the 13 features of Berg-Kirkpatrick et al.'s (2011) method. For each subtree, we try to match it with any of the features. If a match found, we remove that subtree. It is noted that a terminal node (word) remains in a compressed sentence if and only if its parent node is kept in the parse tree after applying the subtree deletion. This process speeds up the compression process and helps us to guarantee that we do not have to match the subtrees whose parent node has already been excluded by the subtree deletion technique. An example of identifying deletable parts from a parse tree is illustrated in Figure 3.2. A subtree (contains “According to the newspaper report”) has been matched with one of Berg-Kirkpatrick et al.'s (2011) deletion features, and it is removed to obtain the compressed sentence.

We performed this compression technique to every sentences of the document set and obtained a set of sentences which includes:

- Original sentences (Compression technique has no effect on those sentences)
- Compressed sentences

[8] For more information, see (Berg-Kircpatricl et al., 2011)

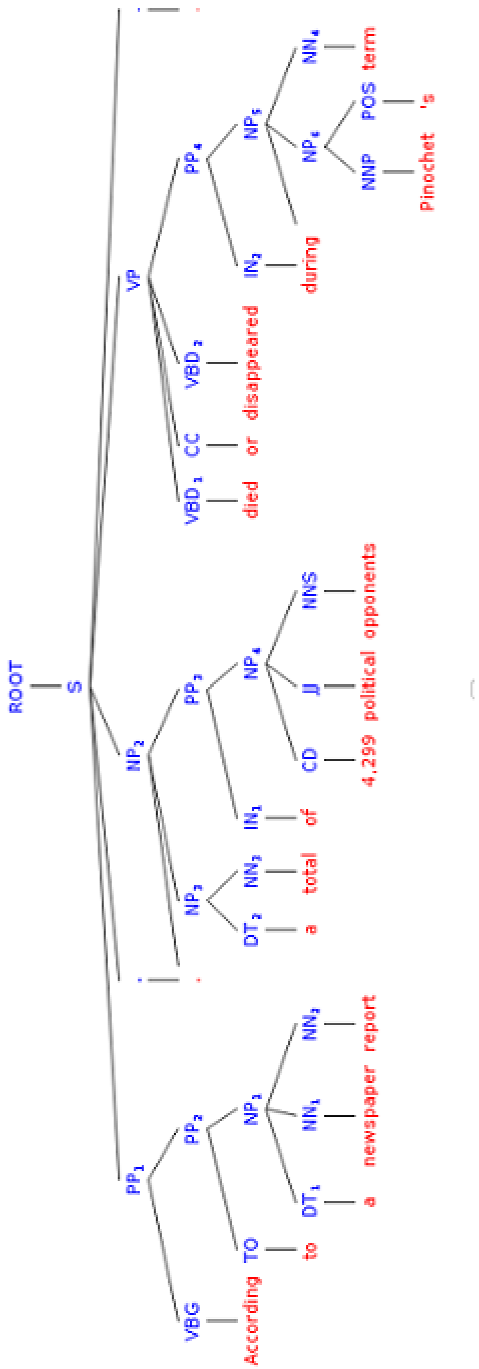

Figure 3.1: Constituency parse tree for the example sentence.

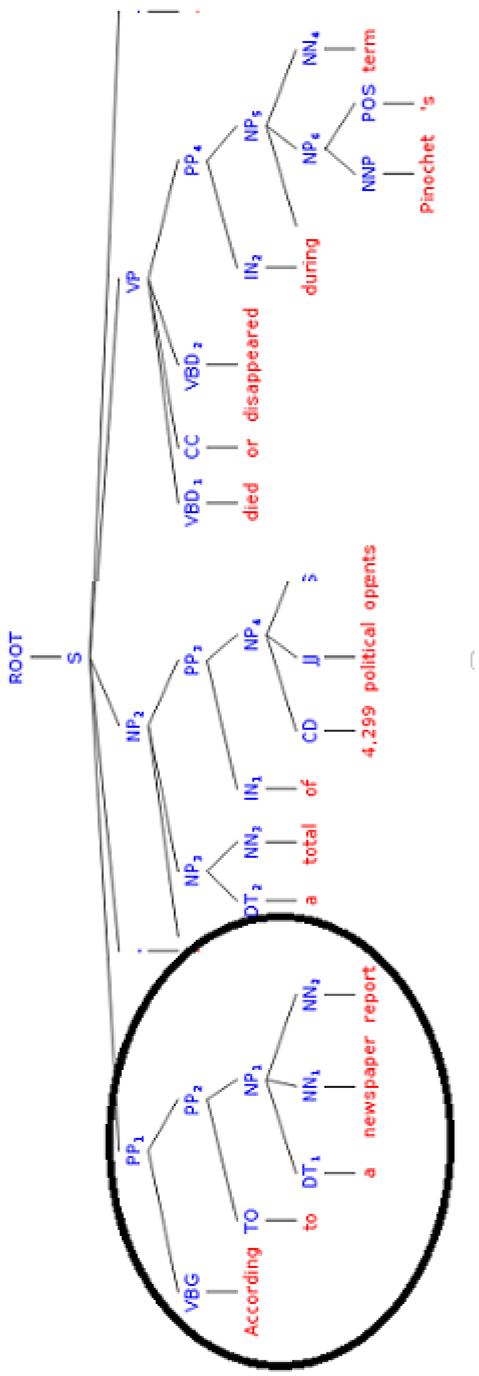

Figure 3.2: Constituency parse tree for the sample sentence.

Table 3.2: Subtree deletion feature taken from Berg-Kirkpatric et al. (2011)

COORD:	Indicates phrase involved in coordination. Four versions of this feature: NP, VP, S, SBAR.
S-Adjunct	Indicates a child of an S, adjunct to and left of the matrix verb. Four version of this feature: CC, PP, ADVP, SBAR
Rel-C	Indicates a relative clause, SBAR modifying a noun
ATTR-C	Indicates a sentence-final attribution clause, e.g. the senator announced Friday.
ATTR-PP	Indicates a PP attribution, e.g. according to the senator.
TEMP-PP	Indicates a temporal PP, e.g. on Friday.
ATTR-NP	Indicates a temporal NP, e.g. Friday.
BIAS	Bias feature, active on all subtree deletions

In addition, we remove the subclause related to the reporting verbs from the sentence using (Chali and Uddin, 2016). Reporting verbs are often used in news documents, though they do not have much significance when we consider the sentence in the summary. Consider the following example sentence:

Cambodia's bickering political parties agreed to a coalition government leaving strongman Hun Sen as sole prime minister, the official said.

In this sentence, we can see a subclause "the official said" does not have much significance in the summary. So, removing this type of unnecessary subclause can improve the summary quality, as well as make space in the fixed length summary for more relevant concepts. For removing the subclause related to the reporting verbs, we consider mostly used reporting verbs such as *said, told, reported*, and *announced* to find out subclause. We used the Stanford Parser (Marneffe and Manning, 2006) to get the dependency parse tree. The dependency parse tree for the above mentioned example sentence is given below.

```
<dependencies type=“basic-dependencies”>
<dep type=“root”>
<governor idx=“0”>ROOT</governor>
<dependent idx=“22”>said</dependent>
</dep>
<dep type=“poss”>
<governor idx=“5”>parties</governor>
<dependent idx=“1”>Cambodia</dependent>
</dep>
<dep type=“possessive”>
<governor idx=“1”>Cambodia</governor>
<dependent idx=“2”>’s</dependent>
</dep>
<dep type=“amod”>
<governor idx=“5”>parties</governor>
<dependent idx=“3”>bichering</dependent>
</dep>
<dep type=“amod”>
<governor idx=“5”>parties</governor>
<dependent idx=“4”>political</dependent>
</dep>
<dep type=“nsubj”>
<governor idx=“6”>agreed</governor>
<dependent idx=“5”>parties</dependent>
</dep>
<dep type=“ccomp”>
<governor idx=“22”>said</governor>
```

```
<dependent idx=“6”>agreed</dependent>
</dep>
<dep type=“prep”>
<governor idx=“6”>agreed</governor>
<dependent idx=“7”>to</dependent>
</dep>
<dep type=“det”>
<governor idx=“10”>government</governor>
<dependent idx=“8”>a</dependent>
</dep>
<dep type=“nn”>
<governor idx=“10”>government</governor>
<dependent idx=“9”>coalition</dependent>
</dep>
<dep type=“pobj”>
<governor idx=“7”>to</governor>
<dependent idx=“10”>government</dependent>
</dep>
<dep type=“xcomp”>
<governor idx=“6”>agreed</governor>
<dependent idx=“11”>leaving</dependent>
</dep>
<dep type=“nn”>
<governor idx=“14”>Sen</governor>
<dependent idx=“12”>strongman</dependent>
</dep>
<dep type=“nn”>
```

```
<governor idx=“14”>Sen</governor>
<dependent idx=“13”>Hun</dependent>
</dep>
<dep type=“dobj”>
<governor idx=“11”>leaving</governor>
<dependent idx=“14”>Sen</dependent>
</dep>
<dep type=“prep”>
<governor idx=“14”>Sen</governor>
<dependent idx=“15”>as</dependent>
</dep>
<dep type=“amod”>
<governor idx=“18”>minister</governor>
<dependent idx=“16”>sole</dependent>
</dep>
<dep type=“amod”>
<governor idx=“18”>minister</governor>
<dependent idx=“17”>prime</dependent>
</dep>
<dep type=“pobj”>
<governor idx=“15”>as</governor>
<dependent idx=“18”>minister</dependent>
</dep>
<dep type=“det”>
<governor idx=“21”>official</governor>
<dependent idx=“20”>the</dependent>
</dep>
```

```
<dep type=“nsubj”>
<governor idx=“22”>said</governor>
<dependent idx=“21”>official</dependent>
</dep>
</dependencies>
```

It is known that a sentence that contains a reporting verb is always being parsed following a fixed simple rule. The rule is that the reporting verb is always the ‘root’ of the dependency tree (Chali and Uddin, 2016). Following this rule, we traverse the tree to find the subclause related to the reporting verb and remove it from the sentence.

Sentence merging

Sentence merging or fusion is another abstraction-based method used in our system. Sentence merging is a technique to create a more informative sentence by merging the information from different source sentences. It can improve the summary quality by reducing redundancy and enhancing information coverage. Also, it is known that, while summarizing the related sentences in the documents, human writers usually merge the important facts into different verb phrases (VPs) about the same entity into a single sentence (Bing et al., 2015). Based on this assumption, we design a sentence merging technique. While in Bing et al. (2015), they took a phrase as the basic linguistic unit and merge phrases to produce a summary, we take a sentence as the basic linguistic unit and merge them to generate new sentences for the summary. Here we illustrate our sentence merging technique with the following example sentences:

The rebels have promised to revitalize the economy by reducing taxes to boost investment.

They will pay civil servants who haven’t seen a paycheck in months or years.

Here, since both the sentences have a coreferent subject (we can observe this by resolving the coreference), it is possible to merge the two sentences by employing only one noun phrase (NP) with the shortest length, as the subject of the newly generated sentence. The details of the sentence merging technique are discussed below.

Compatibility checking while merging:

Before merging two sentences, it is important to check the compatibility relation between them. We used some heuristics to merge two sentences. We only picked the sentences which start with a coreferent subject. The benefit of this heuristic is that it preserves the grammaticality of the newly generated sentences, which is a key challenge in abstractive summarization.

Our system first applies the Stanford Coreference Resolution engine (Lee et al., 2013) on each sentence of a document. From this step, we obtained a set of clusters containing the noun phrases (NP) that refer to the same entity in a document. A new sentence is only generated from two sentences that share a coreferent NP as the subject but have different VPs. We picked the sentences closest to each other for merging and produce the new sentence. The natural order of the sentences has thus been preserved.

The output includes three variants of sentences:

- Newly generated sentences
- Compressed versions of the original sentences
- Original sentences

After this phase, we obtain a cluster of documents containing concise sentences. Now this document set is the input of our document summarization phase.

Grammaticality issue in document shrinking phase:

Grammar quality is an important issue in abstractive summarization. Sometimes sentence compression and fusion can lead to ungrammatical as well as irrelevant sentences. Our system is also no exception to this. From the recent works (Cheung and Penn, 2014;

Bing et al., 2015) we have seen that their systems produced ungrammatical sentences while merging sentence phrases to produce new sentences. Although we minimize the chance of producing ungrammatical sentences by merging only sentences, sometimes it does not contain the relevant information in the merged sentences. The reason is that we have used the Stanford Coreference Resolution engine (Lee et al., 2013), which has certain limitations. Consider the following example sentences:

Anan's brother was angry with his son.

Kofee Anan flew into Libya aboard a special plane from Jerba after receiving clearance from the U.N.

The Stanford coreference engine (Lee et al., 2013) finds Anan's brother and Kofee Anan corefent. So, by merging these two sentences we get a new grammatical sentence but it provides irrelevant information.

In addition, the sentence compression technique that we have used occasionally leads to forming sentences with incomplete information. For example:

Original sentence : Fruit that is grown organically is expensive.

By applying sentence compression, the subordinate clause modifying the noun 'Fruit' is removed and we obtain a sentence like "Fruit is expensive" which contains incomplete information and sometimes misleads the reader.

3.5.2 Document summarization

We consider the text summarization problem as a budgeted submodular function maximization problem similar to the recent works of Lin and Bilmes (2010, 2011). However, our proposed submodular objective function is significantly different from their works, which will be discussed in this section. The main motivation behind this formulation is that the problem can be solved efficiently using an approximation algorithm and the solution is guaranteed to be within a constant factor 0.63 of the optimal solution. Therefore, to take advantage of this efficient algorithm with the specified performance guarantee, we have designed a submodular objective function for document summarization.

Problem definition

Suppose U is the finite set of all textual-units (sentence) in the documents. Our task of summarization is to select a subset $S \subseteq U$ that maximizes the submodular function. Since there is a length constraint in standard summarization tasks (e.g., DUC[9] evaluations), we consider the problem as submodular function maximization with budgeted constraints:

$$\max_{S \subseteq U} \left\{ f(S) : \sum_{i \in S} cost_i \leq B_{max} \right\} \tag{3.1}$$

where $cost_i$ is the non-negative cost of selecting the textual-unit i and B_{max} is the budget. The value of B_{max} could be the number of words or bytes in the summary. $f(S)$ is the submodular objective function that scores the summary quality.

Submodular function for document summarization

We designed a submodular objective function composed of three important objectives for document summarization. These objectives are responsible for measuring the summary's importance, coverage and non-redundancy properties. The proposed objective function is:

$$f(S) = c(S) + \alpha r(S) + \beta h(S) \tag{3.2}$$

where $c(S)$ measures summary's coverage quality, $r(S)$ measures summary's importance quality, $h(S)$ measures summary's non-redundancy quality and α, β are non-negative trade-off coefficients which can be tuned empirically [10].

The linear combination of the submodular functions is submodular (Lin and Bilmes, 2011) and if all the proposed subparts of our objective function are submodular, then the function $f(S)$ is also submodular.

[9] http://www-nlpir.nist.gov/projects/duc/index.html

[10] The values for the coefficients are 1.0, 5.0 for α and β respectively, as found empirically during the experiment.

In the next sections, we discuss the subparts of the designed submodular objective function in more detail.

Importance:

One of the basic requirements of a good summary is that it should contain the most important information across multiple documents. To model this property, we introduced a new monotone nondecreasing submodular function based on the *atomic concept*. In our definition, atomic concepts are the atomic terms that bear significance in a sentence. Our system, therefore, considers only verbs, named-entities, and adjectives as atomic concepts (excluding the stop words). Our proposed submodular function is:

$$r(S) = \sum_{i=1}^{N} \frac{1}{pos(S_i)} \Omega_i . \lambda_{S_i} \tag{3.3}$$

where $\lambda_{S_i} \in \{0,1\}$, $\lambda_{S_i} = 1$ if sentence S_i is in the summary, otherwise $\lambda_{S_i} = 0$. Ω_i is the importance score of sentence S_i and $pos(S_i)$ denotes the position of sentence S_i in the document.

We know the sentences which contain more key concepts in the document are more important. So, we consider that the relevance of the summary is the summation of the importance scores of the sentences in it. Since all the words in a document are not of the same importance, we utilized the Markov random walk model used by (Hong and Nenkova, 2014; Boudin and Morin, 2013; Wan and Yang, 2008; Mihalcea and Tarao, 2004) to score each concept from the document set. Then we scored every sentence based on the weight of the constituent words in the sentence. We only decreased the weight of the constituent concepts when it appears in multiple sentences in the summary. While sentence similarity-based approaches (Lin and Bilmes 2009, 2010, 2011, Takamura and Okumura, 2009b) do not consider the individual word's importance to model the importance property, our proposed submodular function is based on the atomic concept and this model encourages coverage of most of the important concepts across the documents.

Markov random walk model

We used the Markov random walk model for identifying the key concepts across multiple documents. In this model, we construct a graph where the importance of a vertex is recursively computed based on the global information from the graph. The key aspect of the Markov random model is “voting” or “recommendation”. If a vertex is connected to another vertex by an edge, that means it casts a “vote” for it. The importance of vertex depends on two concepts:

1. how many votes the vertex gets (meaning how many vertices are related with the vertex by edges in the graph), and
2. how important those votes are (meaning the importance score of the vertices who cast the votes).

Following (Hong and Nenkova, 2014; Boudin and Morin, 2013; Wan and Yang, 2008; Mihalcea and Tarao, 2004), we constructed a directed weighted graph where vertices are words and edges indicate the co-occurrence relation between the pair of words. We only connected two vertices (words) by an edge if they co-occur within a window of size k. We experimented with different values of k and finally used 4 as the window size. The edge weight is determined by the number of co-occurrence between two words.

Then we applied PageRank (Brin and Page, 1998), a graph-based ranking algorithm for computing the importance score for each node. At first, we initialized the score for each node X_i with a default value. Then the importance score of a word is calculated in an iterative manner until convergence. We used the following equation like the earlier methods (Hong and Nenkova, 2014; Boudin and Morin, 2013; Wan and Yang, 2008; Mihalcea and Tarao, 2004) to calculate the score of each word.

$$S(X_i) = (1-d) + d * \sum_{X_j \in In(X_i)} \frac{w_{ji}}{\sum_{X_k \in Out(X_j)} w_{jk}} S(X_j) \tag{3.4}$$

where $S(X_i)$ is the importance score of vertex (word) X_i, d is the damping factor which we set to 0.85, w_{ji} is the weight of the edge that connects the two vertices i and j, $In(X_i)$ is the set of vertices that point to X_i and $Out(X_j)$ is the set of vertices pointed to by vertex X_j (successors). This equation suggests that a vertex (word) receives a good score if it receives more votes from the neighbouring vertices and if those votes are important.

Content coverage:

A good summary has the capability to cover most of the important aspects of the document (Takamura and Okumura, 2009b). To formulate this concept, we propose a new submodular function, where we emphasize two aspects in scoring the summary: document representativeness and topic word coverage.

The document representativeness of a summary is assessed by considering the fact of how well the summary sentences infer all the core concepts in the document. For scoring the document representativeness of the summary, we utilize the following 'sentence similarity-based approach' based on a "facility location objective" function (Nemhauser et al., 1977).

$$d(S) = \sum_{i \in V} max_{j \in S} sim(i, j) \tag{3.5}$$

where $sim(i, j)$ denotes the sentence similarity between sentences i and j.

We employed two types of sentence similarity - namely cosine similarity and semantic similarity. The main drawback of only relying on cosine similarity is that it does not consider the semantic relation between words when computing sentence similarity. For example: in the case of cosine similarity, it ignores the relation between "Apple" and "Computer" in two different sentences. To overcome this problem, we calculate both semantic similarity along with cosine similarity and take an average of them to achieve a better similarity measure. So our defined similarity measure between two sentences is as follows:

$$sim(i, j) = \frac{\sum_{i,j \in S, i \neq j} cos(i, j) + \sum_{i,j \in S, i \neq j} sem(i, j)}{2} \tag{3.6}$$

where $cos(i, j)$ is the cosine similarity between summary sentences i and j and $sem(i, j)$ is the semantic similarity between summary sentences i and j.

Cosine similarity Measure:

In the process of calculating the cosine similarity, we represent every sentence as a vector of term specific weights (Erkan and Radev, 2004). Two types of parameters are used to calculate the term specific weights. They are term frequency and inverse document frequency. This is also known as term frequency-inverse document frequency (tf-idf) model. After that, we calculated the cosine similarity between a sentence with every other sentence in the document cluster using the formula:

$$cos(i,j) = \frac{\sum_{w \in s_i, s_j} tf_{w,i} \times tf_{w,j} \times idf_w{}^2}{\sqrt{\sum_{w \in s_i} (tf_{w,s_i})^2 (idf_w)^2} \sqrt{\sum_{w \in s_j} (tf_{w,j})^2 (idf_w)^2}} \tag{3.7}$$

where $tf_{w,i}$ and $tf_{w,j}$ are the number of times token w appears in sentence s_i and s_j respectively and idf_w is the inverse document frequency of token w.

Semilar toolkit for semantic similarity:

Our system uses a semantic similarity toolkit SEMILAR (Vasile et al., 2013). This toolkit contains different semantic similarity implementations for text-to-text similarity problems such as word-to-word similarity, phrase-to-phrase similarity, sentence-to-sentence similarity, etc. At first, we used DUC-2003[11] and DUC-2006[12] data[13] for developing LDA models. Then, it uses a LDA-optimal method for calculating semantic similarity between sentences. LDA treats documents as a mixture of topics containing a set of words. The main benefit of using LDA is that a word may belong to more than one topic, and different senses of the word are covered by different topics. Other approaches, such as Latent semantic analysis (LSA), also model the documents into multiple topics, but in LSA, there is only a unique representation for each word (Vasile et al., 2013).

Finally, following equation (3.5), a sentence's eligibility to be included in the sum-

[11] http://duc.nist.gov/duc2003/tasks.html

[12] http://duc.nist.gov/data.html

[13] For generic summarization, we have used DUC-2003 data for LDA topic modeling. For query-focused summarization, DUC-2006 has been used.

mary depends on how similar it is with all the other sentences in the document cluster. A sentence, which is similar to most of the other sentences in the document cluster is regarded as important and should be included in the summary.

However, the major shortcoming of only using the pairwise sentence similarity-based approach (Lin and Bilmes, 2009, 2010, 2011; Takamura and Okumura, 2009b) is that it cannot avoid covering similar concepts in the summary when the topics of the documents are biased. To overcome this problem, we introduced a new concept coverage-based submodular function $g(S)$, giving credit to a concept word only once, which encourages to include more diverse concepts in the summary:

$$g(S) = \eta(S). \sum_{C_k \in S} \sigma(C_k) \tag{3.8}$$

where $\eta(S)$ is the number of distinct concept terms, C_k is the k-th concept term, and $\sigma(C_k)$ is the weight of the concept term C_k, which is calculated using the Markov random walk model.

Finally, combining functions (3.5) and (3.8), we formulate the following submodular objective function $c(S)$ for scoring the content coverage of the summary:

$$c(S) = \Phi. \sum_{i \in V} max_{j \in S} sim(i, j) + \delta.\eta(S). \sum_{C_k \in S} \sigma(C_k) \tag{3.9}$$

Here, Φ and δ are non-negative trade-off coefficients which have been tuned empirically during the experiments[14].

With this formulation, the weakness of pairwise sentence-similarity approach is recovered and more distinct concepts across the documents are covered.

Non-redundancy:

Minimizing redundant information in the summary is another basic requirement in

[14] The values for the coefficients are 2 and 6 for Φ and δ respectively as tuned on DUC - 2003 development set

any summarization task. Most of the past research (Lin and Bilmes, 2010, 2011) minimizes summary redundancy by using a sentence-similarity based approach to penalize the redundant sentences in the summary. It is known that, sentence similarity-based methods do not explicitly consider summary redundancy (Takamura and Okumura, 2009). They always try to cover as many topics as possible. But this implicit control does not always work well because when the topics of the documents are biased, there is high chance of choosing multiple sentences of the same topic in the summary, and ignoring the topic sentences which are less discussed in the document. Another shortcoming of the sentence similarity-based approach is that it does not consider term redundancy in the summary. It is known that a relatively less non-redundant summary contains more distinct concept words than a redundant summary of similar length (Nishino et al. 2013). To overcome these shortcomings, we emphasize two aspects in designing the objective function:

1. Minimizing the similarity among the sentences in the summary.
2. Maximizing the distinct concept term in the summary.

Thus, our proposed submodular function for non-redundancy reward is:

$$h(S) = \sum_{\mathbb{C}_k \in \eta(S)} \sigma(C_k) - \sum_{i,j \in S, i \neq j} sim(i,j) \tag{3.10}$$

where $sim(i,j)$ is the sentence similarity between summary sentence i and j, $\sigma(C_k)$ is the weight of k-th concept term, and $\eta(S)$ is the set of all distinct terms in the summary.

In the function $h(S)$, first our system scores a summary by measuring the weighted sum of the unique concept terms in the summary. Then we penalize the summary redundancy by measuring the sentence similarity among the summary sentences.

Finally, our task is to maximize the proposed submodular function $f(S)$ to produce a relevant, well-covered, and non-redundant summary. The overall architecture of the proposed summarizer is shown in the Figure 3.3.

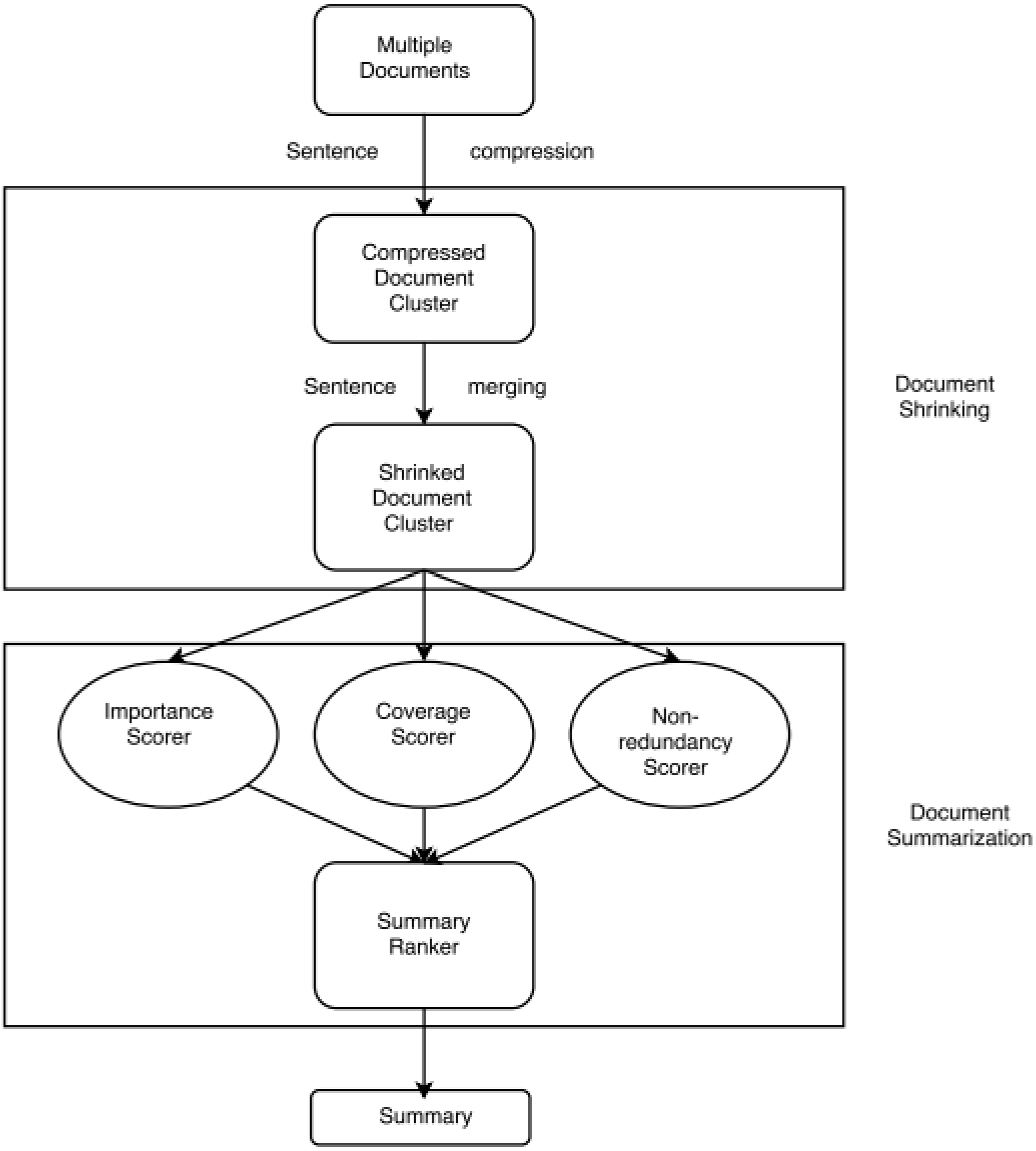

Figure 3.3: Architecture of the proposed document summarizer.

Solving the problem

We consider the problem of document summarization as a budgeted submodular function maximization problem. Maximization of a submodular function is a NP-hard problem. Fortunately the maximization problem of a submodular function under knapsack constraints can be solved near optimally using a greedy search (Lin and Bilmes, 2010). To solve the proposed document summarizer, we implemented the modified greedy algorithm for the submodular function (Lin and Bilmes, 2010) illustrated in Algorithm 1. The algorithm was first proposed by Khuller et al. (1999), but Lin and Bilmes (2010) first used this algorithm for document summarization. The reason behind choosing this algorithm is that a solution is guaranteed to be within a constant factor (1 - 1/e) of the optimal solution when the objective function is submodular. The scoring function $f(s)$ of our proposed summarizer is non-decreasing submodular, therefore, we use this greedy procedure to obtain the near optimal solution.

Algorithm 1 A Greedy algorithm for maximizing the objective function

```
$S \leftarrow \emptyset, M \leftarrow \{1,...,N\}$
while $M \neq 0$ do
    $q \leftarrow argmax_{p\in M}\frac{f(S\cup\{p\})-f(S)}{(c_p)^r}$
    if $\sum_{j\in S} C_j + C_q \leq B_{max}\ \ and\ \ f(S\cup\{q\}) - f(S) \geq 0$ then
        $S \leftarrow S \cup \{q\}$
    end if
    $M \leftarrow M \setminus \{q\}$
end while
$t^* \leftarrow argmax_{t\in\{1,...,N\},c_t\leq B_{max}} f(\{t\})$
if $f(t^*) > f(S)$ then
    return $t^*$
else  return $S$
end if
```

The algorithm selects a sentence q with the largest quality increase of the objective function to the scaled cost. While adding a sentence in the summary, if the length of the sentence violates the budget constraint, the algorithm ignores that sentence and finds another sentence that maximizes the quality increase. Finally, the summary is compared with the within-budget singleton with the largest gain and the algorithm selects the one which has the greatest objective value.

3.5.3 Variants of the Proposed Method

We propose two variants of our system changing the order of document shrinking and summarization. In our first design (Algorithm 2), we perform the document shrinking before the summarization. In the other design (Algorithm 3), the summarization is done first where summaries are generated beyond the budget limit. Then we apply the document shrinking phase to each summary sentence repeatedly until it reaches the summary length. Algorithms 2 and 3 shows these two variants of the proposed method.

Algorithm 2 Algorithm Document shrinking first, then summarization

```
Apply sentence compression to every sentences of the document set
Apply sentence merging technique to the candidate sentences for fusion
Use Alogirhtm 1 to the output document set from the last phase to produce a summary
    of budgeted length
```

Algorithm 3 Algorithm Document summarization first, then shrinking

```
Use Alogirhtm 1 to produce a summary of length beyond the budget
Compress the first sentence
repeat
    Compress the next sentence
    Merge it with the already selected sentences
until summary length $\leq$ budget
```

3.6 Query Focused Summarization

We also propose a summarization model for query-focused summarization. Query-focused summarization is mostly related to web search application where a user query is given and the task is to generate summaries containing query related information as an answer to the query. In doing so, similar to generic summarization, the system must cover all relevant information from different documents. In addition, most importantly the system must gather all query related information in the summary. Therefore, it is really important for the summarization system to consider the query during the summarization process.

The design of our query-focused summarizer is mostly similar to our proposed generic summarizer. The major difference is that for the query-focused summarizer, we have designed a submodular function for query relevance where a submodular function for coverage has been designed for generic summarization. In this section, we briefly discuss the overall methodologies of our proposed query-focused summarizer.

Document processing

We performed the similar document processing steps discussed in Section 3.2. In addition, we have a query expansion step where we expand the given query by adding the synonyms of the query words using Wordnet (Fellbaum, 1998). This extended version of the query is later used for measuring similarity with the sentences of the document cluster.

Proposed Query focused Summarizer

Our proposed query focused summarization system is almost similar to the generic summarizer. We have a similar document shrinking phase. In the document summarization phase, we add a query relevance measure for extracting query-related sentences for the summary. We consider three important measures to select sentences in the query-focused summary.

- Importance (Query independent)
- Query Relevance and Coverage (Query-dependent)
- non-redundancy

For each measure we designed a submodular function, and our final objective function is constructed by adding these three objective functions. Our proposed objective function for query-focused summarization is:

$$f(S) = r(S) + \alpha q(S) + \beta h(S) \tag{3.11}$$

where $r(S)$ measures a summary's importance quality, $q(S)$ measures a summary's query relevance quality, $h(S)$ measures a summary's non-redundancy quality and α, β are non-negative trade-off coefficients which can be tuned empirically [15]. A brief description of these objectives is given in this section.

Importance

We consider an importance function for the query-focused summarization similar to the generic summarization. We adopted a Markov random walk model (Hong and Nenkova, 2014; Boudin and Morin, 2013; Wang and Yang, 2008; Mihalcea and Tarao, 2004) to compute sentence importance. The reason behind formulating this query independent objective function is that even though the summary is query-focused, still requires important sentences in the summary. In addition, if there is more than one sentence of similar length and contain the same amount of query words, it is wise to choose the sentence that has more keywords. This is the motivation of keeping the same importance function in the query-focused summarization task.

Query Relevance and Coverage

This is the most important measure in the query-focused summarization task. While designing the objective function we considered three important aspects:

- how related summary sentences are to the query
- how much query dependent information is covered in the summary
- overall information coverage in the summary

[15] The values for the coefficients are 10.0, 5.0 for α and β respectively, as found empirically during the experiment.

We design the following objective function emphasizing all of the aspects. The first part of the objective function measures the pairwise sentence similarities between the query and the summary. In the second part, our system emphasizes the concept coverage property of the summary by giving credit to the concept words only once. By doing so, the system encourages inclusion of more diverse concept words in the summary. In the third part, our system measures how many query-related terms present in the summary.

$$q(S) = \psi . \sum_{j \in S} Sim(q, s_j) + \zeta . \eta(S) . \sum_{C_k \in S} \sigma(C_k) + \theta . n_{j,q} \tag{3.12}$$

where $Sim(q, s_j)$ is the similarity between summary sentence j and query q. Here, similarity means the average of the cosine similarity and semantic similarity between a summary sentence and the query. The second term denotes the information coverage (see equation 3.8). In the third term, our system measures how many query-related terms are present in the summary. Finally, ψ, ζ and θ are non-negative trade-off coefficients which have been tuned empirically during the experiments[16].

Non-redundancy

We kept the same redundancy objective function which we used for generic summarization. Finally, we applied the modified greedy algorithm (Lin and Bilmes, 2010) to maximize the objective function for obtaining near optimal query-focused summaries.

3.7 Summary

In this chapter, we presented our proposed frameworks for both generic and query-focused multi-document summarization. In both frameworks, we employed two abstraction-based techniques - namely sentence compression and sentence merging for obtaining concise and more informative new sentences. Our proposed summarization system is based

[16] The values for the coefficients are 4, 9, and 2 for ψ, ζ and θ respectively, as tuned on the development set.

on submodularity, where we consider three objectives - namely importance, coverage and non-redundancy - to measure the summary quality. For query-focused summary generation, we consider one more objective query relevance, which measures the summary's relevance with the given query. Finally to obtain the summaries, we employed a greedy algorithm which has $(1 - 1/e)$ performance guarantee when the objective function is submodular. In the next chapter, we show how we evaluate our proposed generic and query-focused summarizers using the ROUGE metrics. Also, we show the comparison between our systems with some of the state-of-the-art systems.

Chapter 4

Experimental Results

4.1 Introduction

In the previous chapter, we presented our proposed generic and query-focused multi-document summarizers. Our proposed generic summarizer takes into account three important measures of a good summary: importance, coverage, and non-redundancy. The query-focused summarizer also considers another important measure query relevance to obtain query related sentences in the summary. Both summarizers incorporate sentence compression and sentence merging to generate summary more concisely. We employed a greedy algorithm with a performance guarantee for the efficient summary generation. In this chapter, we present the experimental results of our proposed summarizer and compare it with the other state-of-the-art summarizers. In the following sections, we describe the task overview, the dataset and the evaluation measures to evaluate the overall quality of the generated summaries.

4.2 Evaluation: Generic Summarization

4.2.1 Task Description

We consider the generic multi-document summarization task defined in Document Understanding Conference (DUC[17]) 2004. Five summarization tasks were defined in that conference. Among them, Task 2 is to generate short multi-document summaries of size ($\leq$ 665 bytes) for each data cluster. To evaluate our generic summarizer, we use the Document

[17] http://duc.nist.gov/

Understanding Conference 2004 (DUC-2004) dataset, which is one of the main benchmarks in the multi-document summarization field. It contains 50 document clusters and each is composed of 10 news wire articles about a given topic from the Associated Press and The New York Times that are published between 1998 to 2000. The dataset also contains multiple human-written summaries which are used for the evaluation of system-generated summaries.

4.2.2 Evaluation measures

We evaluate our system generated summaries using the automatic evaluation toolkit ROUGE [18] (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004). It is the most popular software package to evaluate summaries automatically. It measures the summary quality by comparing it with gold standard summaries created by humans. There are 4 different ROUGE metrics - namely ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. ROUGE-N applies co-occurrence statistics to evaluate a summary, ROUGE-L applies longest common subsequence to compare between two texts, ROUGE-W assigns different weights to consecutive in-sequence matches, and ROUGE-S considers the overlap skip bigrams. Among these four measures, ROUGE-N is the mostly used in multi-document summarization research. It counts the number of overlapping n-grams between the system summary and human written gold standard summaries. We can define ROUGE-N as follows:

$$ROUGE - N = \frac{\sum_{S \in \{R\}} \sum_{g_n \in S} Count_{match}(g_n)}{\sum_{S \in \{R\}} \sum_{g_n \in S} Count(g_n)} \tag{4.1}$$

where n is the length of n-gram, R is the set of reference summaries, $Count(g_n)$ is the number of n-grams and $Count_{match}(g_n)$ denotes the maximum number of n-grams that are both in the candidate summary and a set of reference summaries.

[18] ROUGE package link: http://www.berouge.com

We used ROUGE version 1.5.5 for the summary evaluation. For evaluating our system-generated generic summaries, we used the unigram-based ROUGE measure ROUGE -1 recall and f-measure because they are the main evaluation metric in the DUC-2004 evaluation. In addition, we also used ROUGE -2 (bigram recall). The reason behind choosing both of the metrics is that they have a strong correlation with human judgement.

4.2.3 Experiments

Experiment 1

In this experiment, we investigate the effect of the cost scaling factor, *r* which we have used in Algorithm 1. We performed our experiment with the value of *r* ranging from 0.7 to 2.0. From this experiment, we found the best value for *r* is 1.4, for which we achieved the best ROUGE-1 and ROUGE-2 scores for the summaries. Figures 4.1 and 4.2 show the ROUGE-1 and ROUGE-2 scores when r is between 0.7 to 2.

Experiment 2

In this experiment, we investigate the effect of changing the order of the performing document shrinking phase with the document summarization phase. In our first model, document shrinking is performed before document summarization. In the second model, summarization is performed first and then sentence compression and merging are applied to generate the summaries. Figure 4.3 shows the ROUGE-1 result of this experiment. From the results, we can say the first approach obtains better summaries than the second approach in terms of ROUGE results. The reason is that in the first approach, the system generates more new sentences by merging the sentences which have the same coreferent subject. In addition, by applying sentence compression first, we remove the insignificant part from the sentences. Hence, in both phases, the system reduces the less important part of the candidate sentences and saves space in the summary to include more sentences. In the second approach, the system first extracts the most important sentences, even though the sentences may contain less important information and could be large. Then, the system performs compression on these selected sentences and tries to merge the sentences if possible.

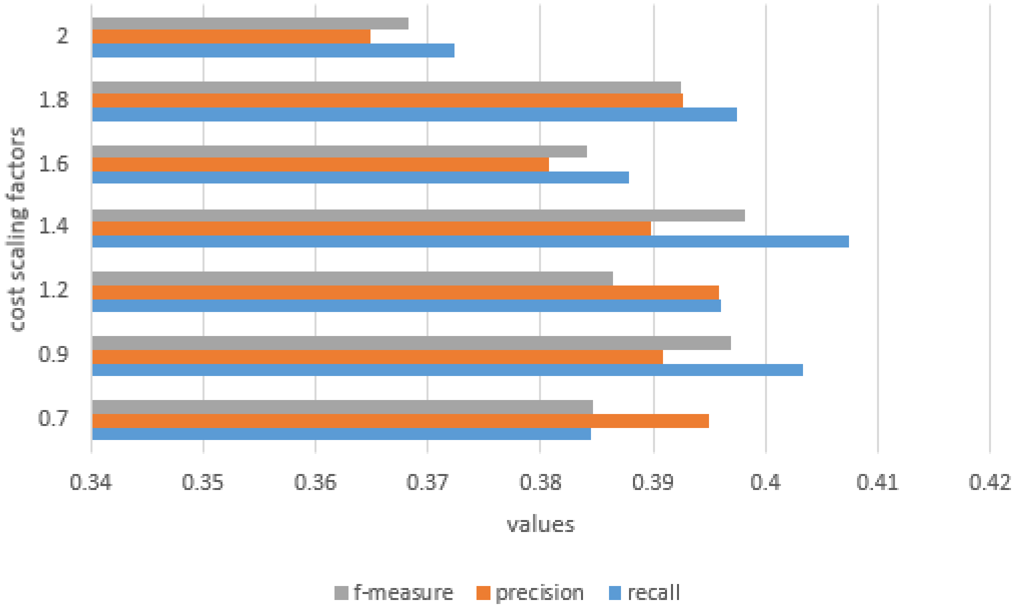

Figure 4.1: ROUGE-1 Recall, Precision and F-measure for different cost scaling factors.

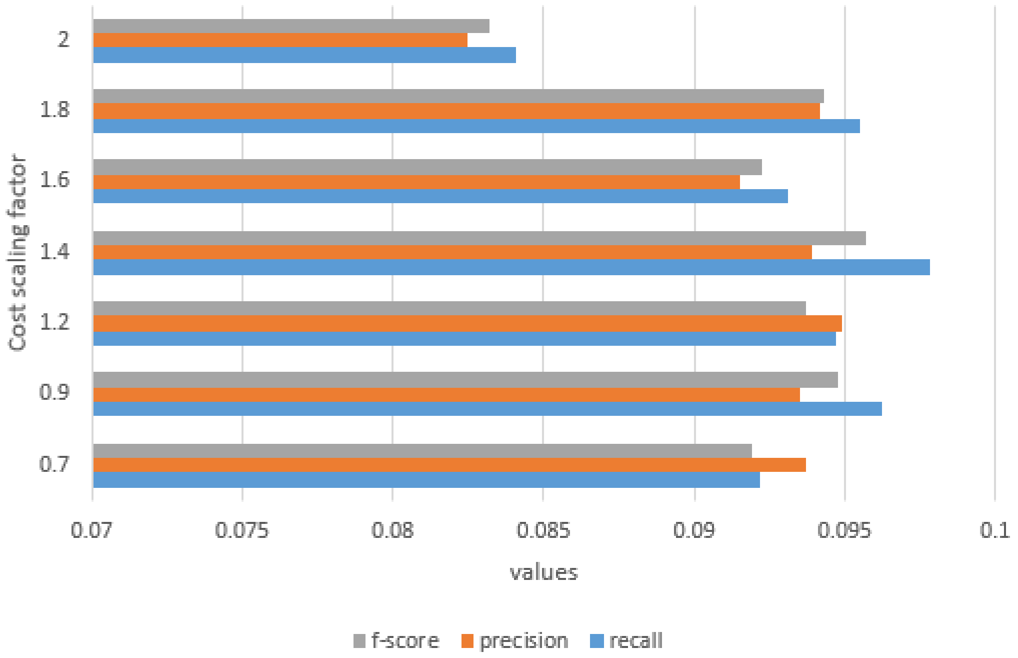

Figure 4.2: ROUGE-2 Recall, Precision and F-measure for different cost scaling factors.

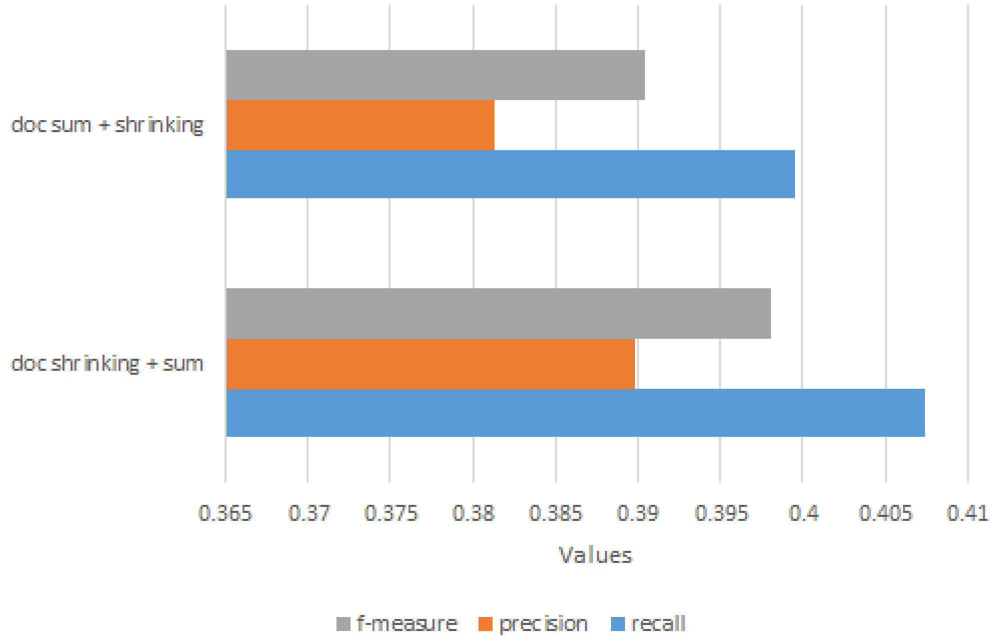

Figure 4.3: ROUGE-1 Recall, Precision and F-measure for two proposed version of summarizers.

But, from the experiment, we found that the merging rate is very low compared to the first approach. The reason is that, among the already selected sentences, it is very rare to have many sentences of the same coreferent subject. Hence, in this approach, the system cannot utilize the power of sentence merging properly.

We also compared the results with other state-of-the-art generic summarization methods such as Lin and Bilmes (2011), Takamura and Okumura (2009), Wang et al. (2009), McDonald (2007), and the best system in DUC-2004 (peer 65). The comparison is shown in Table 4.1 where we report the values of ROUGE[19]-1 recall and f-measures of different approaches. From the table, we can see that our generic multi-document summarizers significantly outperform those systems in both measures.

[19] ROUGE runtime arguments for DUC 2004:
ROUGE -a -c 95 -b 665 -m -n 4 -w 1.2

Table 4.1: ROUGE-1 recall (R1) and F-measure (F1) average results with 95% confident interval on DUC-2004 Dataset. (best result is **bolded**)

Systems	**R-1**	**F1**
Best system in DUC-04 (peer 65)	0.3828	0.3794
G-flow	0.3733	0.3743
Takamura and Okumura(2009)	0.385	-
Lin and Bilmes(2011)	0.3935	0.389
McDonald (2007)	0.362	0.338
Wang et el. (2009)	0.3907	-
Proposed (document shrinking + summarization)	**0.4074**	**0.3981**
Proposed (document summarization + shrinking)	0.3995	0.3904

This result suggests the effectiveness of sentence compression and merging phase in our system. It also shows the effectiveness of using semantic similarity measures to select important sentences in the summary. Moreover, our system also uses a separate redundancy function which helps generate summaries with less redundancy compared to the systems which only concentrate on summary's coverage and relevance. This result also confirms the proposed strategies can improve summary quality.

4.3 Evaluation: Query-focused Summarization

4.3.1 Task Description

We consider the query-focused multi-document summarization task defined in the Document Understanding Conference (DUC) 2007. Two summarization tasks were defined in that conference. They are 1) main task 2) update task. We considered the main task in this thesis. The task was answering complex questions in which the answer should not be simple such as a name, date, quantity, etc. In this task, a topic title and a topic related query are given. Our task is to generate a summary of 250 words that answers the complex question. An example topic is shown below.

<topic>

<num>D0703A </ num>

<title>steps toward introduction of the Euro </title>

<narr>Describe steps taken and worldwide reaction prior to introduction of the Euro on January 1, 1999. Include predictions and expectations reported in the press.

</narr>

</topic>

To evaluate our query-focused summarizer, we used the Document Understanding Conference 2007 (DUC-2007) dataset. The DUC 2007 dataset is made of 45 sets of document clusters and each cluster contain 25 relevant documents. A set of complex questions is also supplied for each cluster. Moreover, the dataset contains multiple human written abstracts which are used in the evaluation of the system summaries.

4.3.2 Evaluation Measures

For evaluating query-focused summaries, we report the widely adopted bigram based ROUGE measure ROUGE -2 recall and f-measure because they are the main evaluation metric from the DUC-2007 evaluation. In addition, we also report ROUGE -1 (unigram) score for the evaluation.

4.3.3 Experiments

Experiment 1

Like the generic summarization evaluation, we run our experiment with the different value of the cost scaling factor to learn its best value. We performed our experiment with the value of r from 0.7 to 2.0. From the experiment, we found r = 1.2, gives us the best ROUGE-2 scores for the summaries. The reason behind adopting ROUGE-2 scores is that they are the main evaluation metric from the DUC-2007 evaluation. Figures 4.4 and 4.5 show the ROUGE-1 and ROUGE-2 score when r is between 0.7 to 2.0.

Experiment 2

In this experiment, like the generic summarization, we also run our experiment changing the order of the document shrinking and document summarization phase in our query-focused summarization evaluation. The results of this experiment are shown in Figures 4.6 and 4.7. From the result, we can again conclude that it is more effective to use the compression and merging operation before sentence extraction for our proposed summarization system.

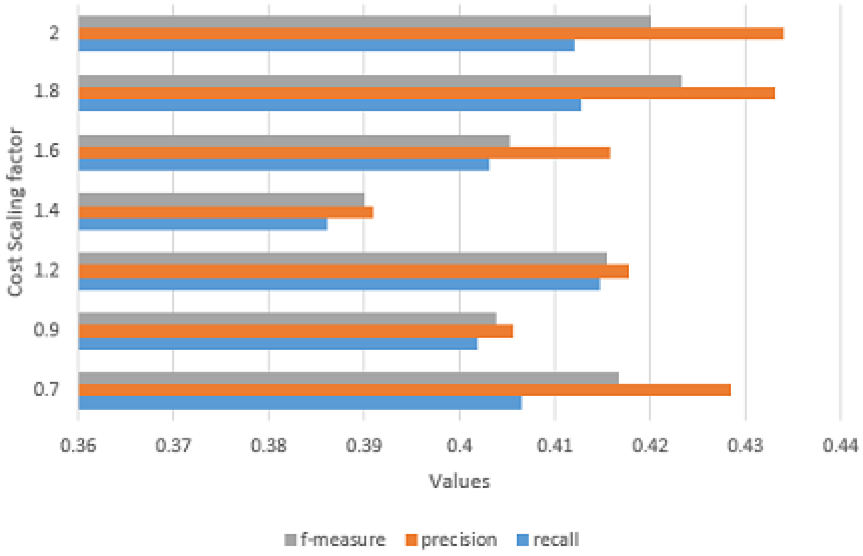

Figure 4.4: ROUGE-1 Recall, Precision and F-measure for different cost scaling factors.

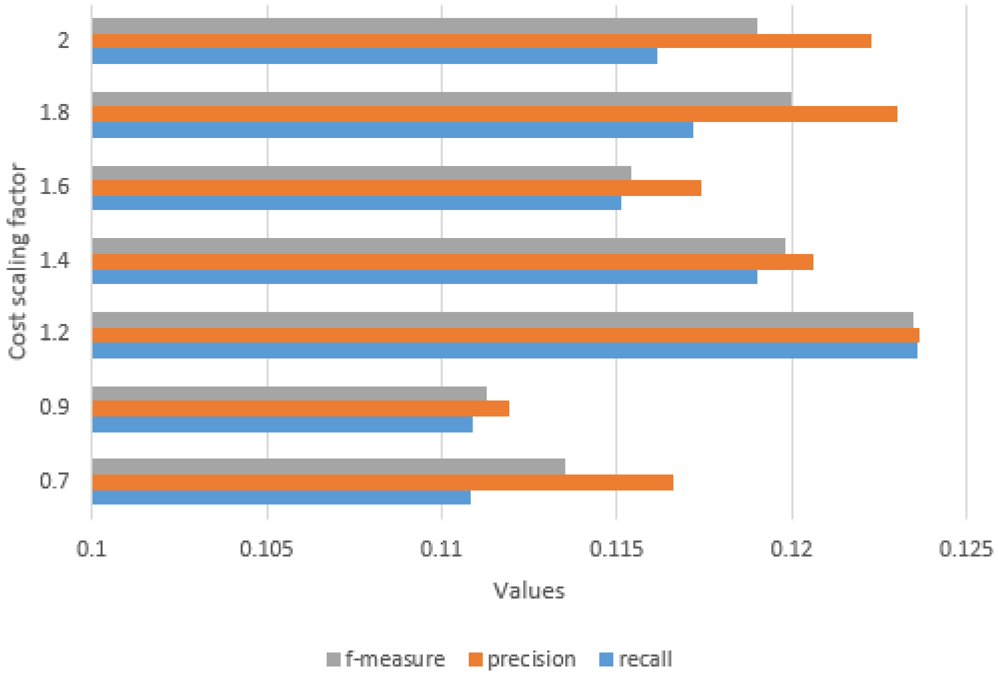

Figure 4.5: ROUGE-2 Recall, Precision and F-measure for different cost scaling factors.

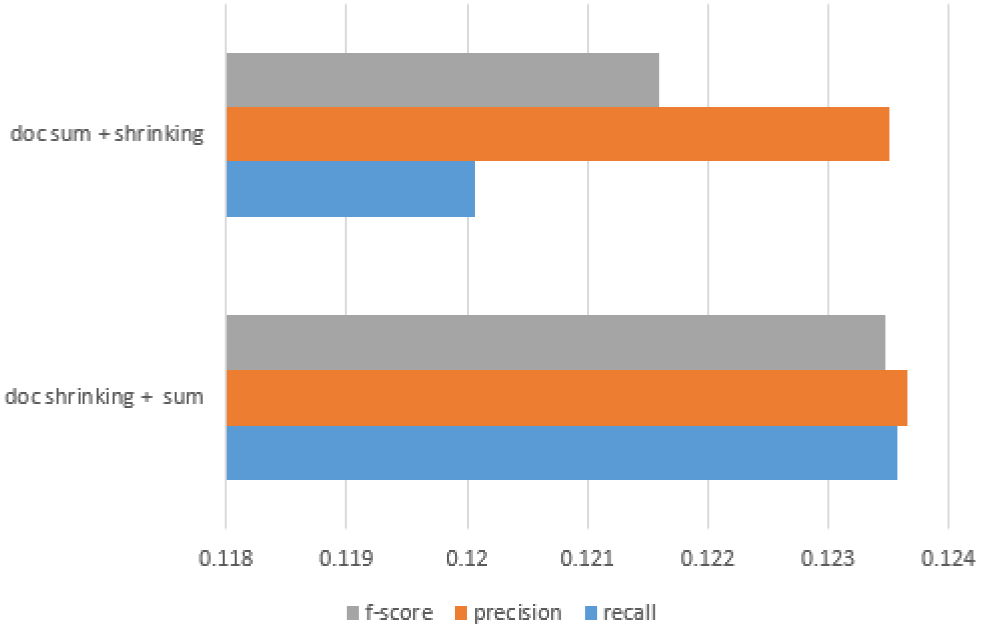

Figure 4.6: ROUGE-2 Recall, Precision and F-measure for two proposed version of summarizers.

We compared the results with other state-of-the-art query summarization methods such as Toutanova (2007), Haghighi and Vanderwande (2009), Celikyilmaz and Hakkani-tur (2010), Lin and Bilmes, (2011), and the best system in DUC-2007 (Pingali and Verma, 2007). Table 4.2 shows the comparison in terms of ROUGE[20] scores between our system and best performing systems. From the table we can say, both of our query-focused multi-document summarizers significantly outperform Totunova (2007), Haghighi and Vanderwende (2009), Celikyilmaz and Hakkani-tur (2010).

Table 4.2: ROUGE-2 recall (R2)and F-measure (F2) results on DUC-2007 Dataset. (best result is **bolded**)

Systems	**R-2**	**F2**
Best system in DUC-07 (peer 15)	**0.1245**	0.1229
Lin and Bilmes(2011)	0.1238	**0.1233**
Toutanova et el. (2007)	0.1189	0.1189
Haghighi and Vanderwende (2009)	0.118	-
proposed (document shrinking + summarization)	0.1235	0.1232
proposed (document summarization + shrinking)	0.12006	0.1216

It also performs similarly to Lin and Bilmes, (2011) and the best system of DUC 2007. It is notable that the best system of DUC 2007 takes the topic title as query and uses the Yahoo search engine to obtain a ranked set of retrieved documents which is used later to calculate the query relevance score (Pingali and Verma, 2007). However, our system is totally unsupervised and does not use any external source for the summary generation.

[20] ROUGE runtime arguments for DUC 2007:
ROUGE -n 2 -x -m -2 4 -u -c 95 -r 1000 -f A-p 0.5-t 0-d

4.4 Summary

In this chapter, we discuss the evaluation results of both the generic and query focused summarizers. At first, we discuss the task and evaluation measures for our system. Then we discussed different experiments. We investigated the cost scaling factor and other tuning factors in several experiments. Also, we examined the order of the document shrinking phase and the document summarization phase and found document shrinking is more effective if it is performed before the document summarization. Finally, we compared our system with other state-of-the-art systems and illustrated how the proposed methods provide improvements over the existing approaches in many occasions.

Chapter 5

Conclusion and Future Works

5.1 Introduction

This thesis focuses on solving different summarization problems. To be more specific, we concentrate on both generic and query-focused multi-document summarization. In this chapter, we will summarize the proposed framework, our contributions, our experimental results to show how our system achieves improved performance over existing summarization systems. At the end of the chapter, we will also provide suggestions to improve the summarization models.

5.2 Summary of the thesis

Document summarization is a well-studied problem in the field of natural language processing. Based on our focus, a summary can be of different types, such as generic summary and query-focused summary. While generic summarization only selects some important sentences to cover the whole collection of documents, a query-focused summary is generated by selecting the sentences which are mostly relevant to the given query. In this thesis, we focus on both of these summarization models. Also, based on the techniques we use, summarization techniques can again be divided into two: extractive and abstractive approach. Extractive approach generates the summary by extracting some relevant source sentences. Here, the system does not require deep language understanding and also we do not have to consider the grammaticality of the sentences in the summary. This method is very popular because of its simplicity and speed. But this approach mostly generates summaries with repeated information. On the other hand, abstractive summarization uses natural language generation to produce human-like summaries. It requires deep language understanding. Though this technique is complex and less popular than the extractive approach, it is possible to produce a more informative and fluent summary. In doing so, researchers often try to modify the candidate sentences by either shortening the sentence or merging information found from different sentences. Different abstraction-based techniques such as sentence compression and sentence fusion are most commonly used to generate abstractive summaries.

The main goal of this thesis is to propose and design summarization systems which can produce a fluent, well-covered, and non-redundant summary. We studied both extractive and abstractive summarization techniques and proposed a summarization approach which is a combination of both. First, our system applies different abstraction-based techniques, such as sentence compression and sentence merging to produce concise and informative sentences. Next, from these concise candidate sentences, we applied our proposed submodularity-based extraction approach to generate summaries. The main motivation of designing such a sentence extraction approach is that it is fast, since a greedy algorithm is used to produce a near optimal summary, which is practical for real world applications. The summary of our model is discussed here.

In this thesis, we proposed a summarization approach that includes two main phases. In the first phase, we applied sentence compression and merging to produce new concise candidate sentences for the summary. This phase is called the document shrinking phase. Sentence compression contributes to our system by removing redundant and less important parts from the sentences. Sentence merging produces more informative sentences by merging sentence fragments coming from different source sentences. We only consider the sentences for merging which start with the same coreferent subject. This heuristic ensures grammaticality of the newly generated sentences. By performing all this, we were able to generate the candidate sentences for the summary, which are either the important sub-part of the source sentences or newly generated informative sentences constituted from different sentence fragments.

The second phase of our approach is document summarization where we propose a sentence extraction approach. This approach is motivated by recent elegant work of Lin and Bilmes (2010,2011), but our work is significantly different from their works. We consider the problem of summarization as a submodular function maximization of budgeted constraints. While generating summaries, our system considers three important measures - namely importance, coverage and non-redundancy - to ensure quality summaries. We designed three submodular functions for these three measures. The importance property of the summary considers how much relevant information present in a summary. We used a Markov random walk model to rank all the key concepts in a document cluster and our model emphasizes to include more key concepts in the summary. The coverage measure ranks the sentences based on how representative they are of the document cluster. It also emphasizes to include more diverse topic-related words in the summary. We can only say, a summary is well-covered if it contains the concept words and also covers the information of most of the sentences in the document cluster. Our third objective function is designed for measuring non-redundancy of the summaries. This metric assigns score to a sentence based on how many distinct concepts it contains and how dissimilar it is with the other summary sentences.

For query-focused summarization, we designed three objectives - namely importance, query relevance and non-redundancy. We kept the same objective functions for measuring the importance and non-redundancy of a summary. However, we introduced a query relevance objective function which is essential for the query-based summarization. It scores a sentence by calculating the pairwise sentence similarities between the query and the sentence. In addition, it considers how many query related words are captured in the summary. Finally, a modified greedy algorithm is applied which has a performance guarantee of (1-1/e) to obtain the summaries.

We evaluated our proposed generic and query focused summarizers following the guidelines of the Document Understanding Conference (DUC). We used the DUC-2004 data for generic summarization evaluation and the DUC-2007 data for query-focused summarization. We performed different experiments to learn different parameters such as cost scaling factors, coefficients of different objective functions, etc. Also, we performed an experiment where we change the order of the document shrinking phase and the document summarization phase to generate summaries. Finally, we compared our system with some of the state-of-the-art systems in terms of ROUGE score. Our generic summarizing system outperforms the well-known systems and our query focused summarizer performs statistically similar to the state-of-the-art systems. This experiments and evaluation clearly demonstrate the effectiveness of our proposed models.

5.3 Future works

Although the results we obtained have shown the effectiveness of the proposed model, it could be further improved in a number of ways:

- Our system uses a sentence merging technique which considers merging only two sentences to obtain a new sentence. In this process, we only used coreference resolution engine to resolve the nouns and pronouns to find the sentences with the same subjects. This approach does not always guarantee relevant new sentences because sometimes two sentences with same subject may contain totally irrelevant and discussed totally different events. We can improve this system with the help of deep linguistic processing. For example, with the current technique, we can apply an event coreference resolution to every sentence. This will allow us to cluster the sentences which describe similar events. From this cluster of sentences, the sentence which have the same subjects can be merged to generate the new sentences. By doing so, it can be possible to get more relevant and informative sentences for the summaries.

  Another approach can be to incorporate both syntactic and semantic sentence similarity measures, along with the coreference resolution technique to cluster relevant candidate sentences for merging. In earlier works, Chali and Joty (2008) proved the effectiveness of using these sentence similarity measures to select the relevant sentences in the summary. Hence, this approach can be a potential solution in achieving more relevant merged sentences for the summary.

- In addition, more investigation is needed to improve the grammar quality of the new sentences.
- A syntactic similarity measure can be incorporated to measure the pairwise sentence similarity. In our proposed approach, we used cosine similarity and LDA-based semantic similarity in our coverage and non-redundancy objective functions. Those similarity measures do not consider word ordering. Hence, it would be interesting to see if applying a syntactic similarity measure could improve the performance of sentence similarity measure as well as the quality of system generated summaries.

Bibliography

[1] Rachit Arora and Balaraman Ravindran. Latent dirichlet allocation and singular value decomposition based multi-document summarization. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 713–718. IEEE, 2008.

[2] Ramakrishna Bairi, Ganesh Ramakrishnan, Rishabh Iyer, and Jeff Bilmes. Multi-topic summarization in dag-structured topic hierarchies via submodular mixtures. In *Proceedings of the Association for Computational Linguistics/Asian Federation of Natural Language Processing (ACL-IJCNLP)*, Beijing, China, 2015.

[3] Rajendra Banjade, Dan Stefanescu, Nobal Niraula, Mihai Lintean, and Vasile Rus. Semilar api 1.0.

[4] Regina Barzilay and Lillian Lee. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 16–23. Association for Computational Linguistics, 2003.

[5] Regina Barzilay and Kathleen R McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005.

[6] Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 481–490. Association for Computational Linguistics, 2011.

[7] Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca J Passonneau. Abstractive multi-document summarization via phrase selection and. *arXiv preprint arXiv:1506.01597*, 2015.

[8] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[9] Allan Borodin, Hyun Chul Lee, and Yuli Ye. Max-sum diversification, monotone submodular functions and dynamic updates. In *Proceedings of the 31st symposium on Principles of Database Systems*, pages 155–166. ACM, 2012.

[10] Florian Boudin and Emmanuel Morin. Keyphrase extraction for n-best reranking in multi-sentence compression. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 298–305, 2013.

[11] Yuri Y Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001.

[12] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336. ACM, 1998.

[13] Asli Celikyilmaz and Dilek Hakkani-Tur. A hybrid hierarchical model for multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 815–824. Association for Computational Linguistics, 2010.

[14] Asli Celikyilmaz, Dilek Hakkani-Tur, and Gokhan Tur. Lda based similarity modeling for question answering. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, pages 1–9. Association for Computational Linguistics, 2010.

[15] Yllias Chali and Sadid A Hasan. On the effectiveness of using sentence compression models for query-focused multi-document summarization. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 457–474, 2012.

[16] Yllias Chali and Sadid A Hasan. Query-focused multi-document summarization: Automatic data annotations and supervised learning approaches. *Natural Language Engineering*, 18(01):109–145, 2012.

[17] Yllias Chali, Sadid A Hasan, and Kaisar Imam. An aspect-driven random walk model for topic-focused multi-document summarization. In *Information Retrieval Technology*, pages 386–397. Springer, 2011.

[18] Yllias Chali and Shafiq R Joty. Improving the performance of the random walk model for answering complex questions. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 9–12. Association for Computational Linguistics, 2008.

[19] Yllias Chali, Shafiq R Joty, and Sadid A Hasan. Complex question answering: unsupervised learning approaches and experiments. *Journal of Artificial Intelligence Research*, pages 1–47, 2009.

[20] Yllias Chali and Mohsin Uddin. Multi-document summarization based on atomic semantic events and their temporal relationships. In *Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings*, pages 366–377, 2016.

[21] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 1, pages 740–750, 2014.

[22] Jackie Chi Kit Cheung and Gerald Penn. Unsupervised sentence enhancement for automatic summarization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 775–786. Association for Computational Linguistics, 2014.

[23] Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. Towards coherent multi-document summarization. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2013)*, 2013.

[24] Janara Christensen, Stephenand Soderland, Gagan Bansal, and Mausam. Hierarchical summarization: Scaling up multi-document summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 902–912. Association for Computational Linguistics, 2014.

[25] James Clarke and Mirella Lapata. Modelling compression with discourse constraints. In *EMNLP-CoNLL*, pages 1–11, 2007.

[26] James Clarke and Mirella Lapata. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, pages 399–429, 2008.

[27] Trevor Anthony Cohn and Mirella Lapata. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, pages 637–674, 2009.

[28] Michael Collins and Nigel Duffy. Convolution kernels for natural language. In *Advances in neural information processing systems*, pages 625–632, 2001.

[29] Anirban Dasgupta, Ravi Kumar, and Sujith Ravi. Summarization through submodularity and dispersion. In *ACL (1)*, pages 1014–1022, 2013.

[30] Hal Daumé III and Daniel Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 305–312. Association for Computational Linguistics, 2006.

[31] Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial OptimizationEureka, You Shrink!*, pages 11–26. Springer, 2003.

[32] Günes Erkan and Dragomir R Radev. Lexrank: graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004.

[33] Katja Filippova. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 322–330. Association for Computational Linguistics, 2010.

[34] Katja Filippova and Michael Strube. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 25–32. Association for Computational Linguistics, 2008.

[35] Katja Filippova and Michael Strube. Tree linearization in english: Improving language model based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 225–228. Association for Computational Linguistics, 2009.

[36] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics*, pages 340–348. Association for Computational Linguistics, 2010.

[37] Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T Ng, and Bita Nejat. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1602–1613. Association for Computational Linguistics, 2014.

[38] Dan Gillick and Benoit Favre. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 10–18. Association for Computational Linguistics, 2009.

[39] Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics, 2009.

[40] Sadid Hasan. *Complex question answering: minimizing the gaps and beyond.* PhD thesis, University of Lethbridge (Canada), 2013.

[41] Kai Hong and Ani Nenkova. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721. Association for Computational Linguistics, 2014.

[42] Rishabh Iyer, Stefanie Jegelka, and Jeff Bilmes. Fast semidifferential-based submodular function optimization. *arXiv preprint arXiv:1308.1006*, 2013.

[43] J Jagadeesh, Prasad Pingali, and Vasudeva Varma. A relevance-based language modeling approach to duc 2005. In *Proceedings of Document Understanding Conferences (along with HLT-EMNLP 2005), Vancouver, Canada*, 2005.

[44] P Pingali J Jagarlamudi and V Varma. Query independent sentence scoring approach to duc 2006. In *In Proceeding of Document Understanding Conference (DUC-2006)*, 2006.

[45] Mikael Kågebäck, Olof Mogren, Nina Tahmasebi, and Devdatt Dubhashi. Extractive summarization using continuous vector space models. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)@ EACL*, pages 31–39. Association for Computational Linguistics, 2014.

[46] Samir Khuller, Anna Moss, and Joseph Seffi Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[47] Kevin Knight and Daniel Marcu. Statistics-based summarization-step one: Sentence compression. *AAAI/IAAI*, 2000:703–710, 2000.

[48] Andreas Krause and Carlos Guestrin. A note on the budgeted maximization of submodular functions. Technical report, CMU-CALD-05-103,Carnegie Mellon University, 2005.

[49] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916, 2013.

[50] Chen Li, Fei Liu, Fuliang Weng, and Yang Liu. Document summarization via guided sentence compression. In *EMNLP*, pages 490–500, 2013.

[51] Chin-Yew Lin. Improving summarization performance by sentence compression: a pilot study. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11*, pages 1–8. Association for Computational Linguistics, 2003.

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

[53] Chin-Yew Lin and Eduard Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 495–501. Association for Computational Linguistics, 2000.

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

[55] Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920. Association for Computational Linguistics, 2010.

[56] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.

[57] Hui Lin, Jeff Bilmes, and Shasha Xie. Graph-based submodular selection for extractive summarization. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, pages 381–386. IEEE, 2009.

[58] Fei Liu and Yang Liu. From extractive to abstractive meeting summaries: Can it be done by sentence compression? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 261–264. Association for Computational Linguistics, 2009.

[59] Inderjeet Mani. *Automatic summarization*, volume 3. John Benjamins Publishing, 2001.

[60] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[61] André FT Martins and Noah A Smith. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 1–9. Association for Computational Linguistics, 2009.

[62] Ryan McDonald. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European conference on IR research, ECIR'07*, pages 557–564. Springer, 2007.

[63] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.

[64] George Miller and Christiane Fellbaum. Wordnet: An electronic lexical database, 1998.

[65] Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978.

[66] Hajime Morita, Tetsuya Sakai, and Manabu Okumura. Query snowball: a co-occurrence-based approach to multi-document summarization for question answering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 223–229. Association for Computational Linguistics, 2011.

[67] Hajime Morita, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Subtree extractive summarization via submodular maximization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1023–1032. Association for Computational Linguistics, 2013.

[68] Hajime Morita, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Ttoku summarization based systems at ntcir-10 1click-2 task. In *NTCIR*, 2013.

[69] Alessandro Moschitti. Making tree kernels practical for natural language learning. In *EACL*, volume 113, page 24, 2006.

[70] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.

[71] Ani Nenkova and Kathleen McKeown. A survey of text summarization techniques. In *Mining Text Data*, pages 43–76. Springer, 2012.

[72] Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In *HLT-NAACL*, volume 7, pages 404–411, 2007.

[73] Emily Pitler, Annie Louis, and Ani Nenkova. Automatic evaluation of linguistic quality in multi-document summarization. In *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*, pages 544–554. Association for Computational Linguistics, 2010.

[74] Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[75] Rahul K Prasad Pingali and Vasudeva Varma. Iiit hyderabad at duc 2007. *Proceedings of DUC 2007*, 2007.

[76] Vahed Qazvinian, Dragomir R Radev, and Arzucan Özgür. Citation summarization through keyphrase extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 895–903. Association for Computational Linguistics, 2010.

[77] Xian Qian and Yang Liu. Fast joint compression and summarization via graph cuts. In *EMNLP*, pages 1492–1502, 2013.

[78] Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tür. Long story short–global unsupervised models for keyphrase based meeting summarization. *Speech Communication*, 52(10):801–815, 2010.

[79] Vasile Rus, Nobal Niraula, and Rajendra Banjade. Similarity measures based on latent dirichlet allocation. In *Computational Linguistics and Intelligent Text Processing*, pages 459–470. Springer, 2013.

[80] Satoshi Sekine and Chikashi Nobata. Sentence extraction with information extraction technique. In *Proceedings of the Document Understanding Conference*, 2001.

[81] Chao Shen and Tao Li. Multi-document summarization via the minimum dominating set. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 984–992. Association for Computational Linguistics, 2010.

[82] Ruben Sipos, Pannaga Shivaswamy, and Thorsten Joachims. Large-margin learning of submodular summarization models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 224–233. Association for Computational Linguistics, 2012.

[83] Hiroya Takamura and Manabu Okumura. Text summarization model based on maximum coverage problem and its variant. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 781–789. Association for Computational Linguistics, 2009.

[84] Kapil Thadani and Kathleen McKeown. Supervised sentence fusion with single-stage inference. In *IJCNLP*, pages 1410–1418, 2013.

[85] Kristina Toutanova, Chris Brockett, Michael Gamon, Jagadeesh Jagarlamudi, Hisami Suzuki, and Lucy Vanderwende. The pythy summarization system: Microsoft research at duc 2007. In *Proc. of DUC*, volume 2007, 2007.

[86] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[87] Mohsin Uddin. Multi-document summarization based on atomic semantic events and their temporal relations. Master's thesis, University of Lethbridge (Canada), 2014.

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

[89] Kristian Woodsend and Mirella Lapata. Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 233–243. Association for Computational Linguistics, 2012.

[90] Wen-tau Yih, Joshua Goodman, Lucy Vanderwende, and Hisami Suzuki. Multi-document summarization by maximizing informative content-words. In *IJCAI*, volume 7, pages 1776–1782, 2007.

[91] Katsumasa Yoshikawa, Tsutomu Hirao, Ryu Iida, and Manabu Okumura. Sentence compression with semantic role constraints. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 349–353. Association for Computational Linguistics, 2012.

Appendix A

Stop Words

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

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

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

Appendix B

Sample system generated summaries

Sample summaries for generic multi-document summarization

In the following, we show an example of our system-generated summary and human-written summary generated from DUC 2004 dataset.

Human-written summary for the document set D31008t

A passerby who found Matthew Shepard's nearly lifeless body tied to a fence outside Laramie, Wyoming at first thought it was a scarecrow. Matthew, an openly gay student at the University of Wyoming, had been kidnapped, brutally beaten and left to die in near freezing temperatures. Two men, Russell Henderson and Aaron McKinney were arrested on charges of kidnapping and attempted first degree murder. Two women, friends of the accused, were charged as accessories after the fact. Seeing this as a hate crime, gay-rights activists nationwide renewed efforts to get the Clinton Administration to pass hate-crime legislation.

Our system-generated generic summary for the document set D31008t

Wyoming is one of 10 states that does not have a hate crime law. Two men, Russell Henderson, 21, and Aaron McKinney, 22, were arraigned on first-degree murder charges. Shepard was born in Casper and attended elementary school, junior high and several years of high school and college, a 21-year-old freshman at the University of Wyoming, had dreamed of working one day for human rights. Matthew Wayne Shepard, the gay student who was beaten in the dead of night, tied to a fence and left to die alone, was mourned at his funeral Friday by 1,000 people, including many who had never met him. The police did not say what the other three did for a living.

Sample summaries for query-focused multi-document summarization

In the following, we show an example of our system-generated summary, baseline summary and human-written summary generated from DUC 2007 dataset.

Query-focused summary of the topic D0720E

Topic title: Oslo Accords

Query: Identify the principles of the Oslo Accord of 1993. Describe what happened in subsequent years in attempts to implement these principles?

Human-written summary

The Oslo Accord of 1993 stipulated that the Israeli Army would pull back from 80% of Hebron in the West Bank by March 28, 1996. A second phase of Israeli withdrawal from the West Bank was to be completed by September 1997. A third and final withdrawal from all of the West Bank would take place before a final peace agreement set for May 4, 1999. The Accords were based on the principle of land to the Palestinians in exchange for peace for the Israelis. Final status talks would include: the status of Jerusalem; the border of the Palestinian entity; the fate of Palestinian refugees; the future of Jewish settlements in the West Bank; the Gaza Strip; and water. The first Israeli withdrawal from the West Bank was delayed after a series of suicide bombing attacks, and again pending the Israeli elections. After the ascent of Netanyahu as Israeli Prime Minister, progress stalled in the face of vigorous Israeli settlement in the West Bank in defiance of the Oslo accords. By 1998 the two sides were still at loggerheads over withdrawal from the West Bank although they should have reached final status talks two years earlier according to the accords. Modification of the Oslo accords was followed by some progress in the Wye Plantation peace agreement in October 1998, but by June 2000 as the Palestinians and Israelis embarked on final status talks, Israel had handed over control of only 40% of the West Bank.

Baseline summary

Arafat was wrong. ET Monday on PBS. The council has 124 members. " Four years ago peace was born in Oslo. Five years later, the rosy picture remains a dream. Covenant also warned that Israel will scuttle the Oslo accords if Arafat declares independence. Announcing these plans, Israel defies all the U.N. resolutions, the Madrid conference principles, the Oslo accords and the will of the international community. Under the Oslo accords, Israel should transfer all remaining West Bank territories to Palestinian control during the third redeployment. Egypt interpreted the Netanyahu proposal as an attempt to annul the land-for-peace principle set by the Madrid conference and the Oslo accords that Israel signed with the Palestinians. Following months of secret talks held in Oslo, the Israeli and Palestinian negotiators reached agreement on August 20 of 1993 on principles of Palestinian self-rule, widely-known later as the Oslo Accords which started the Israeli-Palestinian peace process. The Jewish demonstrators burned a model of Oslo Accords and *tried to enter into the Mosque. That's the only path to peace the Oslo Accords envisioned five years ago, they said. Otherwise, he pointed out, the 1993 Oslo accords could collapse and he would not further implement the Oslo accords if he wins in the May 17 elections. Said holds out little prospect that the Oslo accords will do anything to improve the Palestinians ' plight. When the historic Oslo Accords were signed in Washington today five years ago, the Palestinians were promised land and the Israelis peace.*

(1) Our system-generated query-focused summary

Egypt interpreted the Netanyahu proposal as an attempt to annul the land-for-peace principle set by the Madrid conference and the Oslo accords. Israel and the Palestinians had been expected to sign an agreement over the Israeli withdrawal from the West Bank in Washington Sunday, but nothing happened. The Damascus-based Arab Parliamentary Union denounced Israel's policy of settlement, describing it as a serious challenge to the foundations and principles of the Mideast peace process. Arafat was wrong and has stated many times in the past that under the Oslo principles, Yasser Arafat has the right to declare an independent Palestinian state on that date. 1993 Oslo accords could collapse and he would not further implement the Oslo accords if he wins in the May 17 elections. He explained that the proposed conference is " not to renegotiate the principles of Madrid conference, " rather to call on the parties to implement these principles. Said holds out little prospect that the Oslo accords will do anything to improve the Palestinians ' plight.Oslo Accords have designed a confidence-building mechanism by leaving most of the toughest issues to final status talks, but the depth of recrimination between the two sides has shaken the foundation of the peace treaty. Agreement means everything has been decided on paper and its implementation is more important. Under the Oslo, most of the West Bank should have been handed over to the Palestinians two years ago. Five years later, the rosy picture remains a dream.