2015

Semi-extractive multi-document summarization

Ghiyafeh Davoodi, Fatemeh

Lethbridge, Alta. : University of Lethbridge, Dept. of Mathematics and Computer Science

SEMI-EXTRACTIVE MULTI-DOCUMENT SUMMARIZATION

FATEMEH GHIYAFEH DAVOODI
Bachelor of Science, Amirkabir University of Technology, Iran, 2008
Master of Science, University of Tehran, 2012

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

SEMI-EXTRACTIVE MULTI-DOCUMENT SUMMARIZATION

FATEMEH GHIYAFEH DAVOODI

Date of Defense: August 31, 2015

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. Jackie Rice Chair, Thesis Examination Committee	Associate Professor	Ph.D.

Dedication

Dedicated to my family

Abstract

In this thesis, I design a Maximum Coverage problem with KnaPsack constraint (MCKP) based model for extractive multi-document summarization. The model integrates three measures to detect important sentences including Coverage, rewards sentences in regards to their representative level of the whole document, Relevance, focuses to select sentences that related to the given query, and Compression, rewards concise sentences. To generate a summary, I apply an efficient and scalable greedy algorithm. The algorithm has a near optimal solution when its scoring functions are monotone non-decreasing and submodular. I use DUC 2007 dataset to evaluate our proposed method. Investigating the results using ROUGE package shows improvement over two closely related works. The experimental results illustrates that integrating compression in the MCKP-based model, applying semantic similarity measures to detect Relevance measure and also defining all scoring functions as a monotone submodular function result in having a better performance in generating a summary.

Acknowledgments

I wish to express my sincere gratitude to the following people who support me during my M.Sc. study at the University of Lethbridge.

First and foremost, I would like to express my deepest appreciation to my supervisor Dr. Yllias Chali for his academic support and guidance.

I would like to thank my supervisory committee members, Dr. Wendy Osborn and Dr. John Zhang, for their time and effort spent on my thesis.

I also would like to thank Dr. Hadi Kharaghani and Dr. Howard Cheng, chair and co-chair of the department of mathematics and computer science for providing me such a valuable research environment.

My sincere gratitude to Taylor Berg-Kirkpatrick for kindly providing me valuable information and details of their work on compression features and Giuseppe Pirro for kindly providing me with their API for accessing WordNet.

Special thanks to my husband, Hessam, for his great love and support of my life and study and also to my parents and siblings for their endless support and patience.

Finally, I am also thankful to all my friends and lab mates at University of Lethbridge.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

As the World Wide Web (WWW) is getting bigger and people publish more information on it, users of the WWW have access to, and are overwhekmed with. Considering the volume of relevant information, document summarization has become a must. Document summarization aims at filtering out less informative pieces of documents and only presents the most relevant parts of document(s). Summarizing a vast amount of information is very challenging and more importantly time-consuming, and thus automatic summarization comes as a pragmatic solution. Automatic text summarization is one of the oldest problems which has been investigated in the past half-century by the Natural Language Processing (NLP) and Information Retrieval (IR) communities. Text 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)" (Mani and Maybury, 1999). A summary can be generated by either selecting important sentences of the original text(s) or understanding and rewriting the main idea of the original text(s). It can also be either comprehensive or query specific. In general, the summarization techniques are categorized into different classes based on different criteria as described below:

- Single-document vs. Multi-document summarization: In single-document summarization, a summary is generated from a single document, while in multi-document summarization, a summary is generated from multiple relevant documents.

- Extractive vs. Abstractive summarization: Extractive methods select important sentences and paragraphs from the original document and concatenate them into a shorter form, while abstractive summarization methods understand the original text and rewrite it in fewer words. In an extractive summary, sentences and words are a subset of the original document, while in an abstractive summary, sentences and words may not be in the original document (Mani and Maybury, 1999). Generating an abstract summary with all the features of a good summary is the ultimate goal of automatic text summarization (Genest and Lapalme, 2012). However existing approaches have limited success.

- Query-based vs. Generic summarization: In query-based summarization, a summary is generated with regards to a specific query, while in generic summarization, a summary is generated for general purposes.

According to Mcdonald (2007), three essential criteria are typically considered in selecting a sentence in query-based, extractive, multi-document summarization including: 1) relevance, 2) redundancy and 3) length. The relevance of each sentence is shown its relation to the given query. Sentence redundancy depicts the degree of overlap between the candidate sentence and the generated summary. Finally, length is a constraint on the number of words in the final summary. Coverage is another measure which is considered in some other research (Filatova and Hatzivassiloglou, 2004; Yih et al., 2007; Takamura and Okumura, 2009; Lin et al., 2009) that considers coverage level of a sentence by the document. Sentence compression also has been considered in the process of document summarization (Jing, 2000). Sentence compression can be considered as a word deletion process. It improves the summary quality by removing less relevant words (phrases) from a partly relevant sentence, while keeping the sentence grammatically correct. Thus, the final summary will contain mostly relevant information.

In my thesis, I focus on the query-based, multi-document approaches since any solution for these categories can be easily generalized for generic and single document summarization. In addition, I focus on extractive approaches because: 1) grammatical correctness of linguistic units are preserved at the local level in extractive approaches, 2) problem formulation is quite straightforward (Lin and Bilmes, 2010), and 3) most of recent research focus on extractive approaches. In addition, I employ compression besides commonly used measures in the process of summary generation.

1.2 Problem Statement

The effective aspect of using sentence compression for document summarization has been shown in recent research(Jing, 2000; Knight and Marcu, 2002). It can improve the summary quality by reducing less informative or redundant concepts (words). However, the majority of research concentrates on sentence compression for single-document summarization or generic summarization. Due to the ever increasing volume of data on the web and the necessity for the user to access relevant information based on their current need, having a query-based multi-document summarization is more needed in comparison with single-document or generic summarization.

Maximum Coverage problem with KnaPsack constraint (MCKP) (Filatova and Hatzivassiloglou, 2004) is widely used to model the document summarization problem as it is a good fit for the summarization problem and it is proven to have a great performance (Filatova and Hatzivassiloglou, 2004; Yih et al., 2007; Takamura and Okumura, 2009; Gillick and Favre, 2009; Morita et al., 2011). However, to the best of my knowledge, no research exists that investigates the potential of applying sentence compression in a MCKP model for the task of query-based multi-document summarization. In this thesis, I remodel MCKP by integrating compression into it in the process of generating a summary. To solve the MCKP based summarization problem, many greedy or optimal approaches have been introduced. Optimal approaches are usually expensive or not be practical for a large scale problem (Li et al., 2013). However, they consider the quality of the summary as a whole and mostly generate a summary of higher quality compared to most greedy approaches. On the other hand, greedy approaches do not take summary's quality as a whole into consideration as they generate a summary by applying a heuristic to determine the sentence that looks the best at each step and it may result in generating a lower quality summary. On the other hand, they are not as complex as optimal approaches and can be scaled for a large problem. The work of (Lin et al., 2009) employed a modified greedy approach for the document summarization problem which has a higher quality and maximum scalability because of its greedy nature. However, their approach lacks the compression which may result in generating a summary containing sentences with irrelevant parts. So, in this thesis I cover the compression part by defining it as a monotone submodular function which is compatible with the modified greedy approach.

1.3 Contributions

Although, there has been some research on modeling summarization using MCKP, integration of extraction and compression, or employment of submodulairty in document summarization, my research differs from them in the following aspects:

- Introducing compression into MCKP modeling in the process of generating a summary. This is the first attempt to investigate the potential of applying sentence compression in a MCKP model for the task of summarization to the best of my knowledge.

- Integrating approximation techniques and compression to improve the quality of summarization. The works in (Lin et al., 2009; Lin and Bilmes, 2010, 2011a) take advantage of approximation when their functions have some specific properties (submodularity and monotonicity); However their approach lack the compression part. I integrate compression in their approximation algorithm as another measure to select important sentences to generate a more accurate summary.

- Considering a semantic similarity measure to calculate the relevance of a sentence to a query to better detect the correlation of words. The majority of research uses word-matching based measures, which lack the consideration of semantic relations between words. So, I employ a WordNet based measure to calculate the semantic similarity between a sentence and a query.

- Considering multi-document summarization instead of single document summarization. The majority of research considers only the problem of single-document summarization, while in reality there might be many relevant documents to summarize. Thus, I deal with the more practical scenario of multi-document summarization in my thesis.

1.4 Thesis Organization

The remainder of this thesis is organized as follows:

Chapter 2: I will define automatic document summarization and concepts and also introduce different categorization on it. I will briefly discuss some necessary background, such as WordNet and submodularity. An overview of previous works on automatic document summarization will also be presented.

Chapter 3: I will introduce my semi-extractive document summarization model, which I call it Comp-Rel-MCKP Summarizer. I will explain its preprocessing phase and problem formulation, as well as how I solve the problem and generate a summary using different algorithms.

Chapter 4: I will introduce the Dataset and the evaluation measure. In addition, I will show the result of various experiments to evaluate the effectiveness and efficiency of the proposed model and to compare the model to previous proposed approaches.

Chapter 5: I will conclude the thesis and suggest directions for future research in this area.

Chapter 2

Background on Document Summarization

2.1 Introduction

This section presents some preliminary concepts and definitions for document summarization. In addition, some necessary background for the proposed method including WordNet and WordNet-based similarity measures, which are used to calculate the *Relevance* measure (see Section 3.3.2) and the Submodular function which are used in the proposed approach (see Section 3.3.3) are discussed. Finally, I summarize the existing related approaches for automatic document summarization.

2.2 Background

2.2.1 Document Summarization

Document Summarization is the process of generating a summary from one or more documents. A *summary* is a concise version of a document that contains important information. This process can be done either manually or automatically, and has been considered for many years to reduce the amount of text a user must read. In manual document summarization, a human reads the document carefully and rewrites useful information in fewer words. However, the increasing volume or number of documents makes this a difficult task. To address the problem of manual summarization, Luhn (1958) and Baxendale (1958) introduced automatic summarization in the late 1950s. Automatic summarization assists users by providing a fast and scalable summarizer which can be applied in various domains. The first application of automatic summarization was generating abstracts for literatures (Luhn, 1958) but it was eventually extended to other domains such as summarizing text to be suitable for displaying on hand-held devices (Nagwani and Verma, 2011), or summarizing relevant documents in Search engines, Question Answering and Recommender systems (Wang et al., 2013). Many automatic document summarizers have been introduced since late 1950s. They can be categorized from different perspectives.

The first categorization is based on *"How many documents should be considered in the process of generating a summary?"*. Based on the answer, summarization techniques are categorized into *Single-document* and *Multi-document*. Single-document summarizers consider information from a single document to generate a summary, while multi-document summarizers consider information from multiple documents and generate a single concise summary for all the given documents.

The second categorization is based on *"Whether or not to consider user's need in the process of summarization"* which results in having two categories of *Query-oriented* and *Generic* summarization. Query-oriented summarization methods consider a user's need as a "query" and generate a summary that is related to the given query, while Generic summarization methods generate a summary that has the same variety of topics as the original document(s) and cover the important information of the document(s).

The third categorization is based on *"What is the strategy to select important information to generate a summary?"* which results in having two categories of *Abstractive* and *Extractive summarization*. Abstractive summarizers generate human-like summaries and similar to manual summarization, they need a full understanding of the context and a good ability of rewriting important information into a shorter form. Most abstractive summarizers employ linguistic methods to interpret the document(s) besides advanced language generation techniques (Das and Martins, 2007). Extractive summarizers select important linguistic units of the document(s) and concatenate them to generate a summary. As the proposed method in this thesis is *Extractive*, I focus my discussion on extractive methods only.

Extractive summarizers select important sentences of the document(s) to form a summary. So, one of the main questions is "How to identify a sentence's importance?", which is called *Sentence scoring*. Different measures can be used for scoring a sentence. These measures are as follows:

- *Coverage* evaluates “How much a sentence is representative of the document”. Coverage considers the number of single words, concepts, or n-grams of the document(s), which are covered in the sentence.

- *Relevance* evaluates “How much important content a sentence has (or How important is a sentence)”, which is known as *Importance-based Relevance* and “How relevant is a sentence to the given query in query-oriented summarizer”, which is known as *Query-oriented Relevance*. For importance-based relevance, the position or length of a sentence, or the presence of certain named entities and cue words are some of the features which are considered in the literature. For query-related relevance, word, concept or ngram overlap, longest common subsequence, co-occurrence, and semantic similarity between the sentence and the query are considered in the literature.

- *Redundancy* evaluates “How much a sentence overlaps with the already selected sentences in the summary”. Redundancy can be measured using cosine similarity, syntactic similarity, or semantic similarity measures.

- *Compression* evaluates “How much a sentence is concise and does not contain insignificant information”. Jing (2000) introduced sentence compression in the process of document summarization for the first time as a step toward abstractive summarization. Sentence compression plays an important role in summarization since it allows a summary to have more information by removing insignificant parts. It is usually considered as the number of removed words, concepts, or n-grams. A compression measure is considered at the sentence selection phase in the extractive document summarization, since the chosen sentences may contain insignificant information. There are two main models to employ compression in an extractive document summarization method including the *Pipeline* model and the *Joint* model. In the *Pipeline* model

(Jing, 2000; Knight and Marcu, 2002; Wang et al., 2013), the extraction process is followed or preceeded by the compression process. So, extraction and compression are done in two different phases. But, in the *Joint* model (Daume, 2006; Martins and Smith, 2009; Gillick et al., 2009; Berg-Kirkpatrick et al., 2011; Chali and Hasan, 2012), both extraction and compression are done in a single phase.

- *Diversity* indicates "How much a sentence is different from the selected sentences in the summary". To calculate the diversity value of a sentence, all sentences within the original document(s) are partitioned into different clusters. The diversity measure assigns higher scores to the sentences of a cluster, from which no sentences is already selected in the summary. Lin and Bilmes (2011a) used diversity as a measure for sentence scoring in their proposed summarization approach.

Sentence scoring methods measure at least one of the aforementioned features. They are mainly categorized into three different categories based on how they capture features (Celikyilmaz and Hakkani-Tur, 2010) as follows:

1. *Supervised methods*: These methods need training data to learn the features of a good summary. Then, they assign a score to sentences using the trained features. Sentences are classified as summary or non-summary based on the trained features. Some of the supervised approaches are Bayesian classifier, maximum entropy, conditional random fields (CRF), and skip-chain conditional random fields. Some extractive summarizers that has a supervised sentence scoring are introduced in (Kupiec et al., 1995; Osborne, 2002; Galley, 2006; Yih et al., 2007; Shen et al., 2007; Takamura and Okumura, 2009).

2. *Unsupervised methods*: These methods use some statistical and linguistic features of the document and the dataset to determine the score of each sentence. Some of these features are the location and the statistical features of a term. Some extractive summarizers that use unsupervised sentence scoring are introduced in (Luhn, 1958;

Baxendale, 1958; Marcu, 1997; Schiffman et al., 2002; Daume, 2006; Morita et al., 2011; Lin and Bilmes, 2011a).

3. *Hybrid methods*: These methods combine features from both aforementioned methods to rank sentences. The first hybrid method is introduced in (Martins and Smith, 2009) with others introduced in (Celikyilmaz and Hakkani-Tur, 2010; Berg-Kirkpatrick et al., 2011).

The next step after sentence scoring is, "How to select the best combination of sentences to form a summary". Nenkova and McKeown (2012) categorize different approaches into three main categories: 1) *Best n* approaches, 2) *Greedy-like* approaches, and 3) *Global selection* approaches. In *Best n* approaches, the top *n* sentences having the highest scores while not exceeding the length constraint are chosen to form a summary. In *Greedy* approaches, sentences are selected using an iterative greedy procedure. During each iteration, the scores of sentences are recalculated to reflect their similarity to the already selected sentences in the summary. The sentence not have similar features like the already selected sentences in the summary are dropped from further consideration. Then, a sentence is selected to be added to the summary. In *Global selection* approaches, document summarization is formulated as an optimization problem and tries to solve the problem globally.

As I focus on extractive document summarization in my thesis, I will review some of the proposed extractive document summarizers in Section 2.3.

2.2.2 WordNet

WordNet[1] (Miller et al., 1990) is a lexical database for the English language created and maintained at the Cognitive Science Laboratory of Princeton University. Development of WordNet began in 1985, and it was completed gradually over the years, with the latest version was released in 2006. The purpose of WordNet is twofold: 1) to produce a combination of dictionary and thesaurus and 2) support automatic text analysis. WordNet

[1] Available at https://wordnet.princeton.edu/

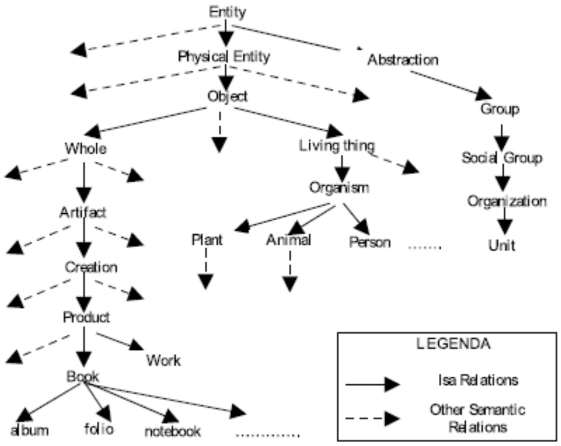


Figure 2.1: A synopsis of noun taxonomy in WordNet (Pirró and Euzenat, 2010)

can also be considered as a lexical ontology. WordNet groups English words into different sets of synonyms which are called synsets (synonym sets). Each synset provides a short and general definition of words which are inside the synset. WordNet also captures the semantic relation between different synsets. The latest version of WordNet contains 155,287 words, which are organized into 117,659 synsets. WordNet consists of four different parts of nouns, verbs, adjectives, and adverbs since they each follow different grammatical rules. Each part is organized in a taxonomy format, and the relations that exist in each part vary from the other parts. As a case in point, for noun, synset *Y* can be holonym of synset *X* if *X* is part of *Y* (e.g. window is part of building). However, the relation holonym does not exist for the other parts (i.e. verbs, adjectives, and adverbs). Figure 2.1 shows a synopsis of the noun taxonomy of WordNet.

As mentioned above, one of the principal goals of WordNet is to support text analysis and find the semantic relation between different concepts and words. Since the development of WordNet, different similarity measures have been proposed. These similarity measures mainly fall into three different, but not necessarily disjoint, categories: Ontology-based (Path-based) approaches, Information Theoretic approaches, and Hybrid approaches.

The first type of similarity measure is *Ontology-based* approach in which the length of the path connecting two concepts which contain the words plays the most important role in calculating the similarity. The first ontology-based approach was proposed by Rada et al. (1989), which considers the distance between two words w_1 and w_2 as the number of links that are needed to attain the Least Common Ancestor (LCA) of concepts c_1 and c_2 containing words w_1 and w_2, respectively. The other approach in this category is introduced in (Pirró and Euzenat, 2010), which is similar to Rada et al.'s similarity measure, but it includes some rules restricting the way concepts are traversed in the taxonomy.

The second similarity measure category is the *Information Theoretic* approach, in which the notion of Information Content (IC) is utilized. This type of similarity measures requires a corpus from which the information content of words is extracted. Resnik (1995) proposed the first approach leveraging IC for the purpose of similarity measure. According to the Resnik's similarity measure, the more probable a concept is of appearing in a corpus, the less informative it would be. In other words, infrequent words have more information to convey. Resnik considers the similarity of two words w_1 and w_2 as the information content of the LCA of concepts c_1 and c_2 (in the taxonomy), which include words w_1 and w_2, respectively.

The last category of similarity measures are *Hybrid* approaches which usually combine multiple information sources. Li et al. (2003) introduced a semantic similarity measure which takes into account the shortest path length, depth, and local density concepts in the taxonomy. The similarity measure used in this thesis to calculate the *Query-oriented Relevance* in Section 3.3.2 is a Hybrid measure called FaITH (Feature and Information THeoretic) proposed by Pirro (2010). This measure takes advantage of two main concepts: ratio-based Tversky's formulation and intrinsic information content. In Tversky's formulation of similarity, which is based on a representation of concepts according to their features, the similarity of two concepts c_1 and c_2 can be calculated by taking into account both common and distinguishing features of c_1 and c_2. As an example, suppose we desire to find the similarity of two concepts "car" and "bicycle" which are descendants of a more general concept "wheeled vehicle". Figure 2.2 illustrates the features of these three concepts.

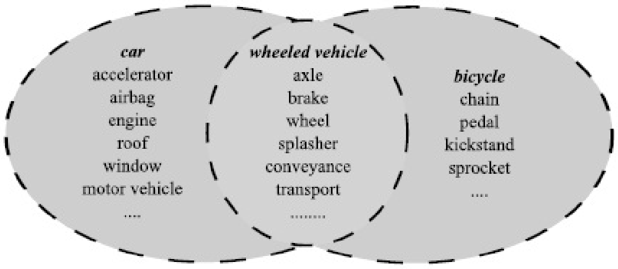


Figure 2.2: Features of concepts *car*, *bicycle*, and *wheeled vehicle* (Pirró and Euzenat, 2010)

The ratio-based Tversky's formulation of similarity of concepts c_1 (car) and c_2 (bicycle) can be represented by the following formula:

$$sim_{tvr_{r}atio}(c_1, c_2) = \frac{F(\psi(c_1) \cap \psi(c_2))}{F(\psi(c_1) \backslash \psi(c_2)) + F(\psi(c_2) \backslash \psi(c_1)) + F(\psi(c_1) \cap \psi(c_2))} \tag{2.1}$$

where F is a function reflecting the salience of a set of features, $\psi(c)$ shows the set of features relevant to concept c, and $F(\psi(c_1) \backslash \psi(c_2))$ means features present in only c_1, and not c_2.

According to the feature-based formulation of similarity in WordNet, F can be replaced by IC in the information theoretic domain. Table 2.1 shows the mapping between feature-based and information theoretic similarity models. Hence, the Formula 2.1 turns into Formula 2.2.

Table 2.1: Mapping between feature-based and information theoretic similarity models (Pirró and Euzenat, 2010)

Description	**Feature-based model**	**Infromation-theoritic model**
Common feature	$F(\psi(c_1) \cap \psi(c_2))$	$IC(msca(c_1, c_2))$
Features of c_1 alone	$F(\psi(c_1) \backslash \psi(c_2))$	$IC(c_1) - IC(msca(c_1, c_2))$
Features of c_2 alone	$F(\psi(c_2) \backslash \psi(c_1))$	$IC(c_2) - IC(msca(c_1, c_2))$

$$sim(c_1, c_2) = \frac{IC(msca(c_1, c_2))}{IC(c_1) + IC(c_2) - IC(msca(c_1, c_2))} \tag{2.2}$$

where msca stands for Most Specific Common Abstraction and $msca(c_1, c_2)$ reflects the information shared by two concepts c_1 and c_2 in an ontology structure. FaITH replaces IC of Equation 2.2 by Extended Information Content (eIC) which is defined as:

$$eIC(c) = iIC(c) + EIC(c) \tag{2.3}$$

where *iIC* is the intrinsic Information Content which is proposed in (Seco et al., 2004) and *EIC* is the Extended Information Content coefficient. *iIC* is defined as follows:

$$iIC(c) = 1 - \frac{log(sub(c) + 1)}{log(max_{con})} \tag{2.4}$$

where the function *sub* returns the number of sub-concepts of a given concept *c*, and *max* is a constant indicating the total number of concepts in the considered taxonomy, which is WordNet here. The coefficient EIC is defined for each concept as follows:

$$EIC(c) = \sum_{j=1}^{m} \frac{\sum_{k=1}^{n} iIC(c_k \in C_{R_j})}{|C_{R_j}|} \tag{2.5}$$

where *m* is the number of all relations where concept *c* is connected to other concepts, *n* is the number of all the concepts at the other end of a particular relation, and C_{R_j} is the set of concepts that have relation to concept c_j.

The final FaITH measure is as follows[2]:

[2] For more information, read (Pirró and Euzenat, 2010)

$$sim_{FaITH}(c_1, c_2) = \frac{eIC(msca(c_1, c_2))}{eIC(c_1) + eIC(c_2) - eIC(msca(c_1, c_2))} \quad (2.6)$$

The FaITH similarity measure reveals a better accuracy in finding the similarity between two concepts compared to other similarity measures, and that is why I have adopted it to calculate the *Relevance* measure (See Section 3.3.2) in my thesis.

2.2.3 Submodularity

Submodularity is widely used in many research areas including game theory, economics, combinatorial optimization, and operations research. Recently it is also considered in NLP research (Lin and Bilmes, 2010, 2011a,b; Morita et al., 2013) since submodular functions can help improving scalability. As I use Submodularity in Section 3.3.3, I explain basic definitions of submodular functions in this section.

Definition

Submodularity is considered as a property of a set of functions (Morita et al., 2013). Let $V = \{v_1, v_2, ..., v_n\}$ be a set of objects, a set function $\mathcal{F} : 2^V \rightarrow \mathbb{R}$ maps subsets of the ground set, $S \subseteq V$, into real values. There are many equivalent definitions of submodularity and two of them are as follows.

Definition 2.1. For any $R, S \subseteq V$, function $\mathcal{F} : 2^V \rightarrow \mathbb{R}$, is *Submodular* if:

$$\mathcal{F}(S \cup R) + \mathcal{F}(S \cap R) \leq \mathcal{F}(S) + \mathcal{F}(R) \quad (2.7)$$

Definition 2.2. For any $R \subseteq S \subseteq V$, and $v \in V$, function $\mathcal{F} : 2^V \rightarrow \mathbb{R}$, is *Submodular* if:

$$\mathcal{F}(S \cup \{v\}) - \mathcal{F}(S) \leq \mathcal{F}(R \cup \{v\}) - \mathcal{F}(R) \quad (2.8)$$

Definition 2.2 is equivalent to the property of diminishing returns which is widely used in economics. It means that a set function $\mathcal{F}$ is submodular if the incremental value of

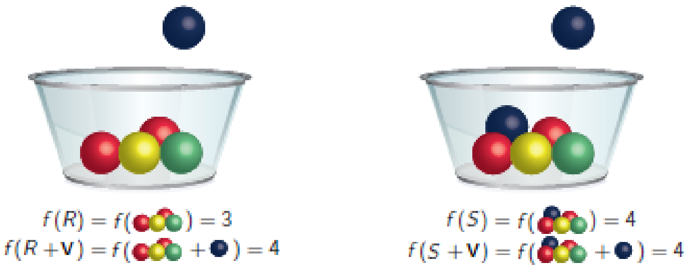


Figure 2.3: Example of submodular function (Lin, 2012)

the function for the superset S, is not greater than the incremental value for the subset R by adding a new element v to both sets. Figure 2.3 shows an example of a submodular function. In this example, function $\mathcal{F}$ counts the number of colors in a container. As it can be seen, the left container has 4 balls with 3 different colors and the right container has 5 balls with 4 different colors. Let us add a new blue ball to both containers. The value of the function $\mathcal{F}$ has an increment of 1 for the left container, however, there is no increment for the right container since it already has a blue ball. So, function $\mathcal{F}$ which counts the number of unique colors in a container is submodular.

Submodular functions can be categorized as *Monotone* and *Non-monotone* and are defined as follows:

Definition 2.3. For any $R \subseteq S \subseteq V$, function $\mathcal{F} : 2^V \to \mathbb{R}$, is *Monotone Submodular* if:

$$\mathcal{F}(R) \leq \mathcal{F}(S) \tag{2.9}$$

Definition 2.4. Any submodular function $\mathcal{F} : 2^V \to \mathbb{R}$, which is not Monotone is *Non-monotone Submodular*.

2.3 Related Works on Automatic Document Summarization

Automatic document summarization was introduced in the late 1950s (Luhn, 1958; Baxendale, 1958). The strategies only considered two measures term frequency (TF) and relative position of words in a sentence to rank sentences and form a summary. However, more automatic document summarizers have been introduced since then, which consider a variety of more advanced features and algorithms in the process of generating a summary. As the number of proposed automatic document summarizers for both *Extractive* and *Abstractive* summarization is quite high, I confine the literature review to solely extractive document summarization approaches, specially I focus on those which consider MCKP for modeling, compression as another measure in generating a summary, or submodularity in defining their scoring functions which are the main focus in this thesis.

Among three different strategies to select sentences to form a summary (discussed in Section 2.2.1), greedy and global selection approaches are more popular in recent years. So, I review some of their related research in this section.

2.3.1 Greedy-like approaches

One of the widely used greedy approaches is *Maximum Marginal Relevance (MMR)* (Carbonell and Goldstein, 1998). This approach considers both the *Relevance* and the *Redundancy* measures in selecting sentences. It gives a penalty to sentences that are similar to the already-chosen sentences in the summary and selects sentences having the highest value of relevance. Erkan et al. (2004) also use MMR to form a summary, but they apply a graph-based method to identify sentence importance. They represent sentences as a graph and apply the concept of eigenvector centrality in the graph to determine sentence centrality. More complicated summarization methods which also use MMR are introduced in (Goldstein et al., 2000; Radev et al., 2004; Dang, 2005).

The summarization method of Schiffman (2002) is another example of a greedy approach. They rank sentences based on some corpus-based features, such as dominant concepts and lead words which are determined using co-occurrence and lead sentences of documents respectively. They consider some features such as the location of a sentence in the document which gives a higher score to the sentence near the beginning of the document.

Then, their method produces a summary by sequentially selecting top-ranked sentences until reaching the desired length.

Filatova and Hatzivassiloglou (2004) also used a greedy algorithm to select important sentences. Their work was the first attempt in which document summarization is formulated as a Maximum Coverage problem with KnaPsack constraint (MCKP). In the MCKP, we are given a set of elements with associated costs and a capacity K. The goal is to find a subset of elements such that the total cost of the subset does not exceed K, and the total weight of elements covered by the selected subset is maximized (Khuller et al., 1999).

To generate a summary, the algorithm selects sentences with the greatest total *Coverage* of words, while implicitly minimizing information overlap within the summary. They believe that the coverage measure simultaneously minimizes redundancy and there is no need to have a seperate measure of redundancy. They show how the coverage measure encompasses redundancy using an example. Consider a case where a document has three concepts A, B, and C and three sentences s_1, s_2, and s_3 as: $s_1 : \{A,B\}$, $s_2 : \{A,C\}$, $s_3 : \{B,C\}$. A good summary should have all three concepts. Using the *Coverage* measure, selecting two sentences is enough to cover all concepts, however, redundancy based measures tend to select all three sentences since any pair of them are partly dissimilar.

Daume (2006) proposed a greedy algorithm called the SEARN algorithm (integrating SEARch and lEARNing) to solve the document summarization problem in which summary is formed incrementally. They concurrently consider a *Relevance* measure which uses some lexical features and language model probabilities of words and sentences, as well as a *Compression* measure that uses the syntactic structure of the sentences.

Yih et al. (2007) also use MCKP to model the summarization problem. However, they consider position related information of a sentence in addition to the *Coverage* measure for sentence scoring, and apply stack decoding to solve it. In their method, they employ supervised learning to learn the probability that a given term in the document will be in the summary.

Takamura and Okumura (2009) also represents document summarization as a MCKP problem, and try to solve the problem both globally and greedily. Their model, which is for generic summarization is based on the two measures of *Coverage* and *Importance*. They also believe that *Redundancy* is implicit in *Coverage*. The *Importance* measure evaluates the relevance level of a sentence to the topic of the document cluster. They employ five different decoding algorithms including 1) a greedy, 2) a greedy algorithm with an approximation factor of $\frac{1}{2}(1-\frac{1}{e})$, 3) a stack decoding, 4) a linear relaxation problem with randomized decoding, and 5) a branch-and bound method. As their result shows, their greedy algorithm outperforms the algorithm proposed in (Filatova and Hatzivassiloglou, 2004). The proposed approach to model the document summarization problem is similar to their approach. However, it differs from their approach since my approach is for query oriented document summarization while their work was for generic summarization and I consider both query-oriented and importance-based features to calculate relevance. In addition, I augment the proposed model with a compression measure which is missing in their model.

Morita et al. (2011) also model the query-based extractive summarization problem based on the MCKP problem and apply a greedy algorithm to solve it. They use an unsupervised method to rank sentences. They enrich the given query using a co-occurrence graph to have a better similarity detection between a query and a sentence.

Recently, submodularity has been used in document summarization (Lin et al., 2009; Lin and Bilmes, 2010, 2011a; Sipos et al., 2012) which results in greedy algorithms with performance guarantees for the summarization process.

Lin et al. (2009) introduce a graph-based document summarization which utilizes submodularity. They build a graph for the document in which vertices indicate sentences of the document and edges indicate a relationship between sentences. A weight, representative of the similarity between sentences, is assigned to each edge. They use the two measures of Coverage and Redundancy to select important sentences and apply a greedy algorithm to generate a summary in which a constant cost is considered for all sentences.

Lin and Bilmes (2010) also propose another document summarizer using a submodular function which is a generalization of their previous work (Lin et al., 2009). They formulate summarization as a submodular function consisting two measures of Redundancy and Coverage. In their previous work, they consider an identical cost for all sentences. However in (Lin and Bilmes, 2010) the cost of sentences varies based on their lengths. They propose a greedy algorithm with a $(1-\frac{1}{\sqrt{e}})$ performance guarantee which is inspired by the greedy algorithm introduced in (Khuller et al., 1999) for the budgeted maximum coverage problem. This greedy algorithm needs the scoring function that is monotone and submodular.

Lin and Bilmes (2011a) improve upon their previous works using two measures of *Relevance* and *Diversity* to rank sentences. They apply their modified greedy algorithm proposed in (Lin and Bilmes, 2010) to generate a summary. They believe that *Diversity* is a good replacement for the widely used measure of *Redundancy*, since *Redundancy* violates the monotonicity of the objective function. In their objective function, *Diversity* assigns higher score to the sentences of a cluster, from which no sentences is already selected in the summary. Employing submodular functions in the proposed model is inspired by Lin and Bilmes's work (2011a). However, the scoring functions are different. They use Diversity as a replacement measure for Redundancy and a different Relevance measure, while my proposed approach is based on MCKP and I use three measures of coverage which implicitly contain redundancy, relevance and compression to score sentences.

Sipos et al. (2012) also proposed a supervised approach to learn submodular scoring functions. They consider two submodular measures of *Redundancy* and *Coverage* in their extractive document summarization. They use the same *Redundancy* measure as introduced in (Lin and Bilmes, 2010) which considers inter-sentence similarity.

Dasgupta et al. (2013) also work on integration of submodular functions in document summarization. Their work is a generalization of (Lin and Bilmes, 2011a) which considers a combination of submodular and non-submodular functions. They employ a *Redundancy* measure which is non-submodular and apply a different greedy algorithm which has a 1/4-approximation factor.

2.3.2 Global selection approaches

Global selection is another strategy to generate a summary. One of the first global selection approaches is introduced by McDonald (2007) which is an improvement to Goldstein's method (2000) by considering the MMR as a knapsack problem. They employ a dynamic programming algorithm and Integer Linear Program (ILP) to maximize the optimality of the generated summary. As their result shows, their approach improves the performance of the summary.

Gillick et al. (2008; 2009) also introduced a global selection approach using a concept-based ILP approach to generate a summary. They consider three measures of *Relevance*, *Coverage*, and *Compression* in their work. Their method generates a compressed version of sentences and considers them besides original sentences during the sentence selection process to form a summary. In (Gillick et al., 2009), they remove temporal and modifiers expressions using semantic role labeling to generate a compressed sentence.

Martins and Smith (2009) proposed a joint model for integrating extractive document summarization and compression as a global optimization problem using ILP. Their method used a supervised tree-based model for sentence *Compression*, and *Relevance* and *Redundancy* measures for extraction.

Berg-Kirkpatrick et al. (2011) proposed a supervised tree-based approach to compress and extract sentences simultaneously. They model their joint approach as an ILP in which objective function is based on *Coverage* and *Compression* which is based on subtree deletion model (in terms of number of cut choices in the parse tree of a sentence). They used an approximate solver for their compressive-extractive summarizer to generate a summary. Firstly, they extract a subset of sentences with a total length of *M* or less. Then, they generate a summary for the selected subset of sentences using their joint model and ILP. Their work is another closely related work to the proposed approach, however, they do not employ submodular functions and a greedy algorithm, but instead use Integer Linear Programming. Their proposed method is for generic summarization in which they use supervised learning, while the proposed method is for query-based summarization and I use unsupervised learning.

Chali and Hasan's document summarization method (2012) is also a global selection approach. They also used ILP to formulate query-based multi-document summarization, but they considered three measures of *Relevance*, *Redundancy*, and *Compression* in their work. They also investigated the result of three models of *ComFirst*, in which compression is performed on all sentences first, before compressed sentences are selected to form a summary, 2) *SumFirst*, in which important sentences are selected first, before the selected sentences are compressed, and 3) *Combined*, in which compression and extraction are performed jointly. As their result shows, their *Combined* model outperforms the two other models. They also investigate the result of three different compression models using language models with lexical and syntactic constraints, topic signature modeling function, and semantic role constraints.

Among the different measures of *Coverage*, *Relevance*, *Redundancy*, and *Compression* discussed in Section 2.2.1, *Compression* has been considered by many strategies since it can really affect the quality of generated summaries. Even though, some strategies solely concentrate on how to compress a sentence without considering it as a step in a summarization framework. So, in the following I review some of the main researches on compression.

Grefenstette (1998) proposed the first attempt to employ sentence compression in automatic summarization. They use a rule-base approach to summarize audio for the blind.

Sentence compression methods use different modeling paradigms such as the noisy-channel model (Knight and Marcu, 2002; Turner and Charniak, 2005; Galley and McKeown, 2007; Zajic et al., 2007), decision-tree learning (Knight and Marcu, 2002), constituency or dependency parse tree (Jing, 2000; McDonald, 2006; Berg-Kirkpatrick et al., 2011), margin-based learning (McDonald, 2006; Berg-Kirkpatrick et al., 2011), and language model (Clarke and Lapata, 2008).

Jing (2000) was one of the first to apply sentence compression for automatic text summarization. They propose a supervised method that uses syntactic information of the sentence. They employ an English Slot Grammar (ESG) parser (McCord, 1989) to build the sentence parse tree, apply some grammatical checking,context information, syntactic knowledge, and statistics derived from a corpus to determine insignificant phrases of each sentence.

Knight and Marcue (2002) proposed two models of compression using a noisy channel and a decision tree to compress a sentence. The noisy channel model consists of three models of source model, channel model, and decoder. In the source model, the grammatical correctness probability of each string *s* in the sentence, P(s), is calculated. P(s) shows how likely a string *s* can be considered in the compressed version of the sentence. The channel model calculates the probability of $P(S|s)$ which shows how likely the string *s* can be converted/expanded to the sentence *S* by adding additional words. The decoder component of their model, searches for the string *s* in the sentence *S* that maximizes $P(s) * P(S|s)$. In their decision-tree model, they use a parse tree of the sentence and apply a shift-reduce paradigm to compress the sentence. They show that a decision-tree model is more flexible compared to noisy channel model.

Tuner and Charniak (2005) improve upon the work of (Knight and Marcu, 2002) by proposing both an unsupervised and a semi-supervised modified noisy channel model. Galley and McKeown (2007) and Zajic et al. (2007) also propose sentence compression approaches based on the noisy channel model introduced in (Knight and Marcu, 2002).

McDonald (2006) proposed a discriminative approach to compress sentences. They build a phrase-based, dependency parse tree for each sentence and use some soft syntactic features of the parse trees to compress sentences.

Clarke and Lapata (2008) formulate sentence compression as an optimization problem and apply integer linear programming (ILP) to generate compressed sentences. They use a language model to determine unimportant n-grams within the sentences and employ some hand-crafted constraints to ensure the grammatical correctness of the compressed sentences.

2.4 Summary

In this chapter, I explain document summarization and the different categorizations on it. Extractive summarization steps are discussed in detail as it is the focus in this thesis. Some related works are then reviewed. Even though several approaches have been proposed for extractive document summarization, there is no work on the integration of MCKP and compression to the best of my knowledge. In addition, no research is done on employing compression in submodular based models which has a good performance. The following chapter describes the proposed approach to cope with the aforementioned shortcomings in document summarization. More specifically, the approach provides an improvement over the works discussed in Sections 2.3.

I employ compression in the proposed document summarization strategy using the Joint model discussed in Section 2.2.1, since Pipeline model may fail to find an optimal solution, regardless of which operation, compression or extraction, is performed first.

Chapter 3

Semi-Extractive Document Summarizations

3.1 Introduction

The problem of extractive document summarization has been studied extensively because of the ever increasing volume of relevant available information. Despite the popularity of extractive approaches, they are still suffering from a fundamental problem which is "whether or not to select a lengthy sentence with partly relevant information" (Wang et al., 2013). Including a relevant but lengthy sentence in the summary may result in excluding other relevant sentences due to the space limit. However, excluding a relevant sentence from the summary may result in missing relevant information.

Sentence compression has been considered as a good remedy to the aforementioned problem (Jing, 2000). It can improve the summary quality by reducing less informative or redundant concepts or words. There exist two principle approaches to extract the most important sentences, greedy approaches and optimal approaches[3]. Many approaches have focused on using greedy algorithms to extract the important sentences due to their simplicity and speed. However, the major limitation of the available approaches is that little attention is given to integrating approximation techniques and compression to improve the quality of summarization. This motivates us to propose a compression-based extractive (semi-extractive) summarizer that integrates compression in the latest approximation algorithm for document summarization.

To model document summarization, I use the Maximum Coverage KnaPsack (MCKP) problem since the task is to select a subset of sentences in extractive document summarization. In addition, based on the definition of summary, it should be of a specific length.

[3] These approaches were discussed in Section 2.2.1.

Thus, it can be easily mapped to a knapsack problem.

In this chapter, I introduce the proposed semi-extractive document summarizer. The proposed approach is not fully extractive, since fully extractive methods either include a sentence in the summary entirely or completely exclude it from the summary. However, in the proposed method, a sentence can be partially included in the summary.

3.2 Notations and Definitions

In this section, I first introduce the most important notations used in the automatic document summarization (see Table 3.1).

Table 3.1: List of notations in automatic document summarization

notation	**explanation**
D	each document in the dataset
S	summary
s	linguistic unit
e	conceptual unit
K	summary length
c	cost of a linguistic unit
t	(sub)tree
$p(s)$	parse tree

3.3 Proposed Summarizer

Based on the categorization explained in Section 2.2.1, I only focus on a query-based and multi-document approach, in this thesis as any solution from this category can be easily generalized for generic and single document summarization. In addition, I only consider extractive approaches because 1) grammatical correctness of linguistic units are preserved at the local level in extractive approaches, 2) problem formulation is quite straightforward (Lin and Bilmes, 2010), and 3) most of recent research focus on extractive approaches.

The proposed document summarization can be divided into the following steps: *Preprocessing*, *Problem formulation*, and *Solving the problem*. Each step is further discussed in the following sections.

3.3.1 Preprocessing

Preprocessing plays a key role in efficient summary generation. Preprocessing first decomposes a document D into several linguistic units s_i. I consider sentences as linguistic units of a document. Not only does it ensure the grammatical correctness, but also prevents the impracticality in detecting other linguistic units (Takamura and Okumura, 2009). Sentences which contain quotations are discarded in the process of decomposing each document to its sentences to improve summary recall since they are not appropriate for summary (Gillick et al., 2009). So, the documents are shown as $D = \{s_1, ..., s_{|D|}\}$. Next, each sentence s_i is decomposed to some conceptual units e_{ij} (i.e. $s_i = \{e_{i1}, ..., e_{i|s_i|}\}$). Conceptual units of a sentence can be its words, named entities, syntactic subtrees or semantic relations. Some research has been carried out on determining conceptual units (Hovy et al., 2006). However, their usefulness has not been proven for document summarization. Most research (including this thesis) use *Words* as the conceptual units due to its simplicity (Takamura and Okumura, 2009). Besides, inappropriate concept extraction can be biased towards sentence ranking and therefore results in a low quality summary (Gillick and Favre, 2009). Next, I apply the Porter Stemmer (Porter, 1980) to represent each word by its stem [4] using the Porter Stemmer (Porter, 1980). Stemming, which is a process to reduce all words with the same root to a common form is widely used in NLP and the document summarization fields. Stemming is useful because different forms of a word may be used in documents. It finds a common form for all different forms of a word which helps us to better detect the

[4] Normally terms originating from a common root or stem have similar meaning. For instance, words INTERSECT, INTERSECTING, INTERSECTED, INTERSECTION, and INTERSECTIONS all have root INTERSECT, and the process of finding word's roots is called Stemming.

correleation between words. The next step is detecting stop words[5]. However, unlike most research, the detected stop words are not removed from the documents in my proposed approach because removing stop words from a sentence affects its grammatical correctness. I, however, detect the stop words in the proposed approach in order to bypass them in the sentence scoring process.

3.3.2 Problem formulation

MCKP is a good fit for the document summarization problem since it is used to determine the word coverage easily (or the concept coverage in general). So, in this thesis, the proposed document summarization method is based on MCKP. The goal of document summarization as MCKP is to cover as many conceptual units as possible using only a small number of sentences. However, in query-based summarization methods, the relevance of the generated summary to a given query and the compression ratio of the summary are also important. So, in the proposed summarization technique which I call ***Comp-Rel-MCKP*** document summarization, three measures of ***Coverage***, ***Relevance***, and ***Compression*** are considered. The goal of *Comp-Rel-MCKP* document summarization is to generate a summary while maximizing the value of all three measures. In the next sections, each measure will be discussed in more detail.

Coverage

The coverage measure represents coverage level of conceptual units by any given set of textual units (Filatova and Hatzivassiloglou, 2004). It evaluates how a sentence is representative of a document. Two different coverage functions introduced in (Takamura and Okumura, 2009) are used for measuring the coverage level of a summary and a sentence. The *Coverage* function for summary, $Cov(S)$, shows how the generated summary S covers D and is defined as follows:

[5] Stop word detection means identifying words such as ”And” and ”Or” which do not convey any special concept in a sentence. There is no predefined list of stop words in English, but the stop word list which I have used in this thesis consists of 319 words which is shown in Appendix A.

$$Cov(S) = \sum_j z_j \qquad \forall j, e_j \in S \tag{3.1}$$

where z_j is 1 when word e_j is covered in the summary S, and 0 otherwise and j is the number of words in the summary S. $Cov(S)$, considers the number of unique words in the summary as the coverage score.

The *coverage* function for sentence s_i, $Cov(s_i)$ is similar to $Cov(S)$, but it considers summary S in its measurement. That is, $Cov(s_i)$ measures the number of unique words in the sentence s_i which are not covered by the already selected sentences in the summary S. $Cov(s_i)$ is defined below:

$$Cov(s_i) = \sum_j z_j \qquad \forall j, e_j \in s_i \text{and} e_j \notin S \tag{3.2}$$

The aforementioned *Coverage* functions have the advantage of implicitly encompassing the notion of redundancy because redundant sentences cover fewer words.

Relevance

The *relevance* measure represents the importance of a given set of textual units as well as its correlation with a given query. The relevance function is considered as a combination of a set of query-oriented and importance-oriented measures. The query-oriented measures consider the similarity between a sentence and the given query while the importance-oriented measures calculate the importance of a sentence in a given document (Chali and Hasan, 2012; Edmundson, 1969; Sekine and Nobata, 2001) regardless of the query. Relevance function for a summary or a sentence is calculated in the same way and relevance function at summary level is defined as follows:

$$Rel(S) = \sum_i sim(s_i, q) + imp(s_i)$$
$$\forall i, s_i \in S \tag{3.3}$$

where $sim(s_i, q)$ and $imp(s_i)$ are the query-oriented and importance-oriented features respectively, each reveal similarity of sentence s_i to the given query q and the importance of the sentence s_i regardless of considering the query q respectively.

Relevance function at the sentence level is defined as follows:

$$Rel(s_i) = sim(s_i, q) + imp(s_i) \tag{3.4}$$

Many works use vocabulary matching between the query q and the sentence s_i to calculate $sim(s_i, q)$. They consider the number of words that the sentence s_i overlaps with query q as their similarity score (Lin and Bilmes, 2011a). Vocabulary matching similarity measure is easy to calculate. However, it fails to detect any semantic similarity between words. For example consider the following query and sentence:

Query: “Describe the state of teaching art and music in public schools around the world. Indicate problems, progress and failures.”

Sentence: “The nonprofit foundation is dedicated to restoring music programs in schools nationwide and raising public awareness about the importance of music education.”

Existing matching based similarity measures fail to detect a high similarity score between this query and sentence as semantic relations between vocabularies are ignored in these measures. For example, they ignore the relation between “teaching” in the query and “education” in the sentence. One remedy to this problem is to exploit WordNet-based measures which consider the semantic relations between words in order to calculate the similarity between sentence s_i and query q. To calculate the semantic similarity between sentence s_i and query q, $sim(s_i, q)$, both sentence s_i and query q are represented as a vector of words (bag of words) after tokenization and the stop word removal process. The way I find the semantic similarity between two vectors of words, representing the sentence and query is inspired by the maximum weighted matching problem in a bipartite graph.Then, the semantic similarity is calculated as follows.

Figure 3.1 illustrates the vectors of words representing sentence s_i and query q. For each word e_{s_ij}, $(j = 1, ..., m)$ in the vector of words s_i, I find the semantic similarity of e_{s_ij} to all words e_{qk}, $(k = 1, ..., n)$ in the vector of words q using the FaITH similarity measure of WordNet (discussed in Section 2.2.2). I then assign word e_{qk} of the query to the word e_{sj} of the sentence which has the highest similarity. As the word types of e_{s_ij} and e_{qk} are unknown, I have to look them up in all four parts of WordNet (noun, verb, adjective, and adverb parts), and assign the highest similarity among the four similarity values that I come up with as the similarity of e_{s_ij} and e_{qk}. Therefore, I have $4mn$ semantic similarity look up in total. There are cases in which one or both of e_{s_ij} and e_{qk} does not exist in WordNet. In such cases, I assume that their similarity is zero. After assigning all of the words in the s_i to a word in the q and calculating the pair similarities, the total semantic similarity of s_i and q is the result of summing up all the pair similarities divided by the total number of words in the vectors of words s_i and q, $(m+n)$.

To calculate $imp(s_i)$ which represents the importance of a sentence, I combine the TF-IDF measure and the inverse position of the sentence. The TF-IDF measure is widely used in the information retrieval and document summarization areas and presents a good estimation of the importance of a textual unit. In addition, the position of a sentence is also used as a good indicator of importance in document summarization, as early sentences tend to be more important (Gillick and Favre, 2009; Chali and Hasan, 2012). Thus, the importance of sentence s_i, $imp(s_i)$, is defined as:

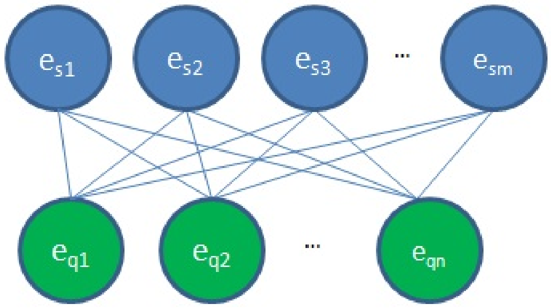


Figure 3.1: Vectors of words representing s_i and q

$$imp(s_i) = \alpha \sum_{j \in s_i} TF - IDF(e_{kj}) + \beta \frac{1}{Pos(s_i)}$$
$$\forall j, e_j \in s_i \text{and} s_i \in S \tag{3.5}$$

where TF and IDF is the *Term Frequency* and *Inverse Document Frequency* for word e_j respectively, within its original document d_k and $Pos(s_i)$ indicates the position of the sentence s_i within its original document d_k. For example, the first sentence in a document has a position of 1, the second sentence has a position of 2, and so on. $TF - IDF(e_{kj})$, weight of the word e_j in the document d_k, is calculated as:

$$TF - IDF(e_{kj}) = tf(e_{kj}) * log_{10} \frac{N}{df(e_j)} \tag{3.6}$$

where $tf(e_{kj})$ is the frequency of the word e_j in the document d_k, N is the number of documents in the corpus[6], and $df(e_j)$ is the number of documents in the corpus which contain word e_j. This formula determines how relevant a given word is in a particular document. Words that are used in a single or a small group of documents tend to have higher $TF - IDF$ value than common words that are used in most of the documents.

[6] Duc 2007 data set is considered as the corpus

Compression

Sentence compression plays a key role in summary generation as it reduces wasted space wasting and enhances the chance of including more relevant information. So, compression is considered as another measure in the process of generating a summary in this thesis. Ideally, it should detect redundant or insignificant parts of a sentence, while keeping the important parts such that the readability and correctness of the sentence are preserved. Sentence compression is considered as a challenging task which should deal with all of these parameters. Consider the following sentence[7] as a candidate to be added to a summary.

"Thousands of jobless demonstrated across France on Monday, to press the Socialist-led government for a bigger increase in unemployment benefits and a Christmas bonus, according to the official way of accounting unemployment."

In this example, the insignificant parts are underlined. The compressed sentence is "Thousands of jobless demonstrated across France, to press the Socialist-led government for a bigger increase in unemployment benefits and a Christmas bonus." As it can be seen, removing the insignificant part of the original sentence preserve the significant information, readability, and grammatical correctness of the sentence. In the process of sentence compression which is viewed as a word deletion process, I remove deletable parts of a sentence using Berg's compression method (2011). I define the compression function at the summary level as:

$$\begin{aligned} &Comp(S) = \sum_{i} d_{ij} \\ &\forall i, s_i \in S, \forall d_{ij} \in DS(D) \end{aligned} \tag{3.7}$$

[7] This sentence is from DUC 2007, topic D0701A

where d_{ij} denotes a constant which is 1 if word e_j is deleted from sentence s_i, 0 otherwise, and $DS(D)$ contains insignificant parts of the entire document.

Compression function at the sentence level is defined as:

$$\begin{gathered} Comp(s_i) = \sum d_j \\ \forall d_j \in DS(s_i) \end{gathered} \tag{3.8}$$

where $DS(s_i)$ contains insignificant parts of the sentence s_i.

Considering all three described measures, the described goal and summary length constraints, the objective function is defined as:

$$\begin{gathered} Maximize F(S) = \alpha Cov(S) + \beta Rel(S) + \gamma Comp(S) \\ = \alpha \sum_j z_j + \beta \sum_i (sim(s_i, q) + imp(s_i)) x_i + \gamma \sum_i (d_{ij}) x_i \\ \text{subject to} \sum_i c_i x_i \leq K, \sum_i a_{ij} x_i \geq z_j \\ \forall i, x_i \in \{0,1\}, \forall j, z_j \in \{0,1\} \\ \forall d_{ij} \in DS(D) \end{gathered} \tag{3.9}$$

where α, β, and γ are scaling factors for *Coverage*, *Relevance*, and *Compression* respectively. The variable x_i is set to 1 if sentence s_i is selected, and 0 otherwise. The summary length, K, as introduced in 3.1, is measured as the number of words. In addition, let the constant a_{ij} is 1 if sentence s_i contains word e_j , and 0 otherwise. The word e_j is considered as covered when at least one sentence containing e_j is selected in the summary. The variable c_i is the cost of selecting s_i, which is measured as the number of words in s_i. $DS(D)$ contains insignificant parts of the entire document. In other words, for each sentence, it contains some parts that I can remove while keeping its grammatical correctness and informative parts. As I discussed before, the constantd_{ij} is 1 if word e_j is deleted from sentence

s_i, and 0 otherwise. So, the goal is to find a binary assignment on x_i with the best value for the measures such that the summary length is at most K.

To calculate $DS(D)$, the first step is to generate a constituency parse tree[8], $p(s)$, for each sentence using the Berkeley parser (Petrov and Klein, 2007). Figure3.2 illustrates a constituency parse tree for a sample sentence.

The second step in order to find deletable parts of sentence s_i, is to detect subtrees in the parse tree of $p(s_i)$ as a set of $T = \{t_{1i}, t_{2i}, ..., t_{mi}\}$, where m is number of possible subtrees in $p(s_i)$. Then, method of Berg-Kirkpatrick et al. (2011) is applied on each subtree in T, to detect deletable parts. Berg-Kirkpatrick et al.(2011) introduced thirteen features which were trained on the TAC dataset using human annotated data sets of extracted and compressed sentences. Table 3.2 explains their subtree deletion features[9]. Finding the features on the generated parse tree of a sentence will result in determining deletable subtrees or sometimes the entire parse tree.

[8] A constituency parse tree of constituency grammars (= phrase structure grammars) distinguish between terminal and non-terminal nodes. The interior nodes are labeled by non-terminal categories of the grammar, while the leaf nodes are labeled by terminal categories

[9] For more information, see (Berg-Kirkpatrick et al., 2011)

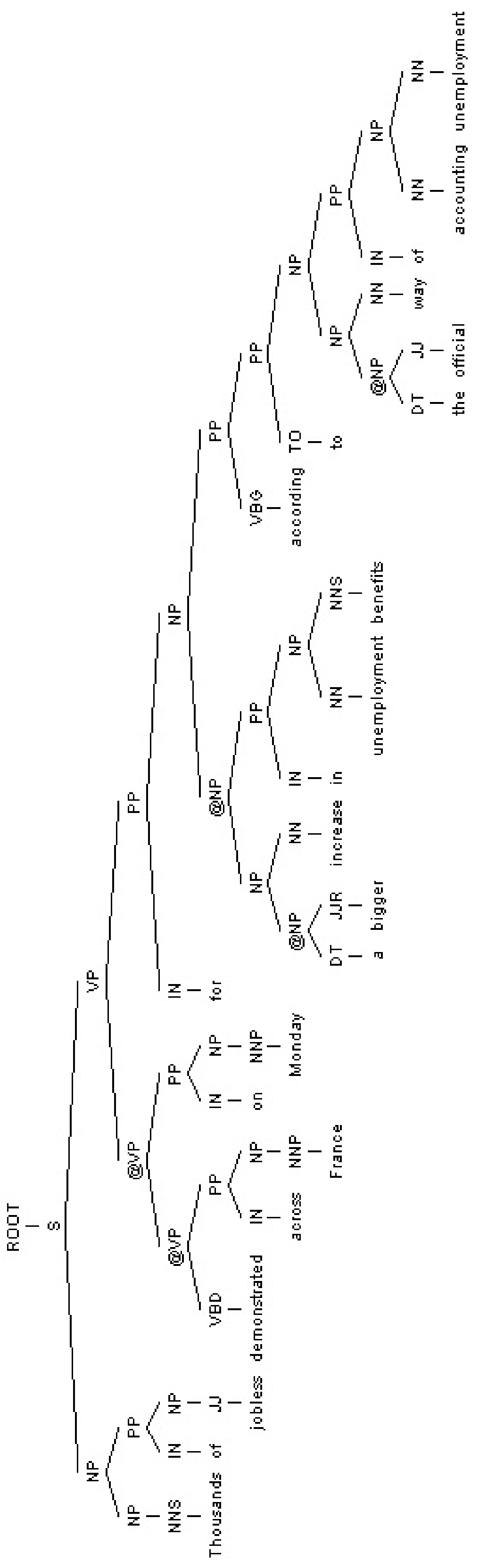


Figure 3.2: Constituency parse tree for a sample sentence

Table 3.2: Subtree deletion features (Berg-Kirkpatrick et al., 2011)

COORD:	Indicates phrases involved in coordination. Four version 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 the proposed summarization model, I decide on whether or not to prune each subtree in the constituency parse tree of a sentence. To represent the compressed summary S, let p_{s_i} be a constituency parse tree for sentence s_i and $S = (e_j : j \in p_{s_i}, s_i \in D)$ be a vector of indicators of non-terminal nodes in each parse tree as a representative of the summary. Word e_j of sentence s_i will be in the summary, if and only if, its parent node in the parse tree has been presented in the summary. It means that any node of e_j may have $e_j = 1$ only if its parent $\pi(j)$ has $e_{\pi(j)=1}$. This constraint helps us to guarantee that only subtree may be deleted and it speeds up the compression process since the proposed system stops investigating all subtrees of a (sub) tree t_j if the system decides not to include t_j in the summary. Figure 3.3 illustrates a sample compressed sentence using parse tree.

The overall architecture of *Comp-Rel-MCKP* document summarizer is shown in Figure 3.4.

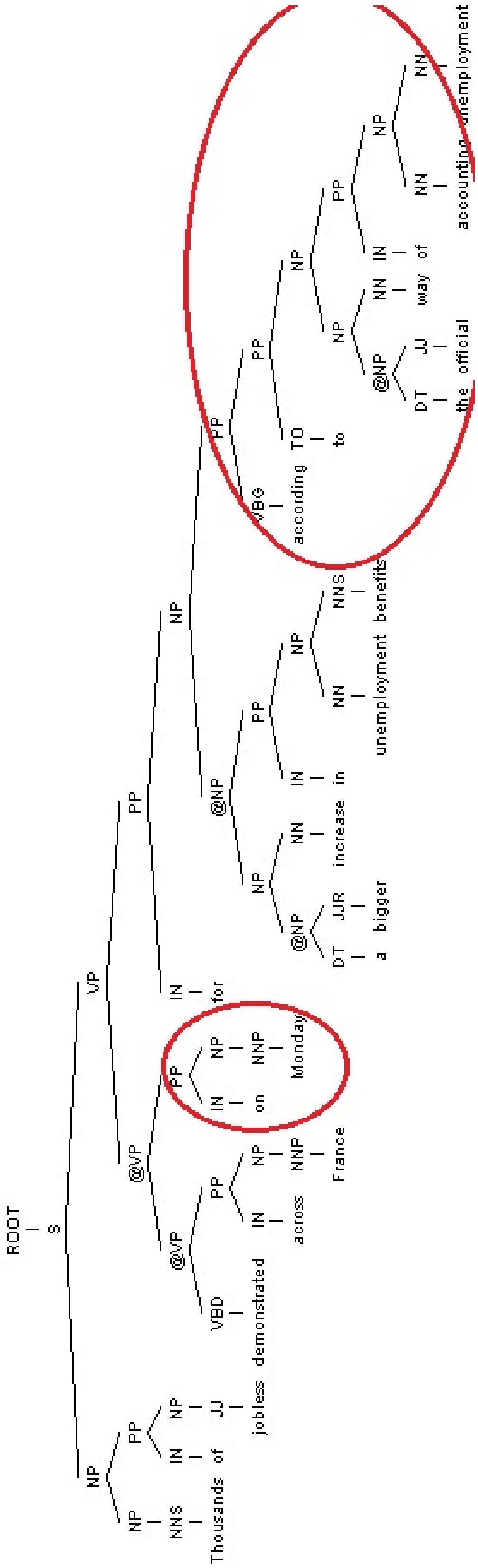


Figure 3.3: Subtree deletion for a sample sentence

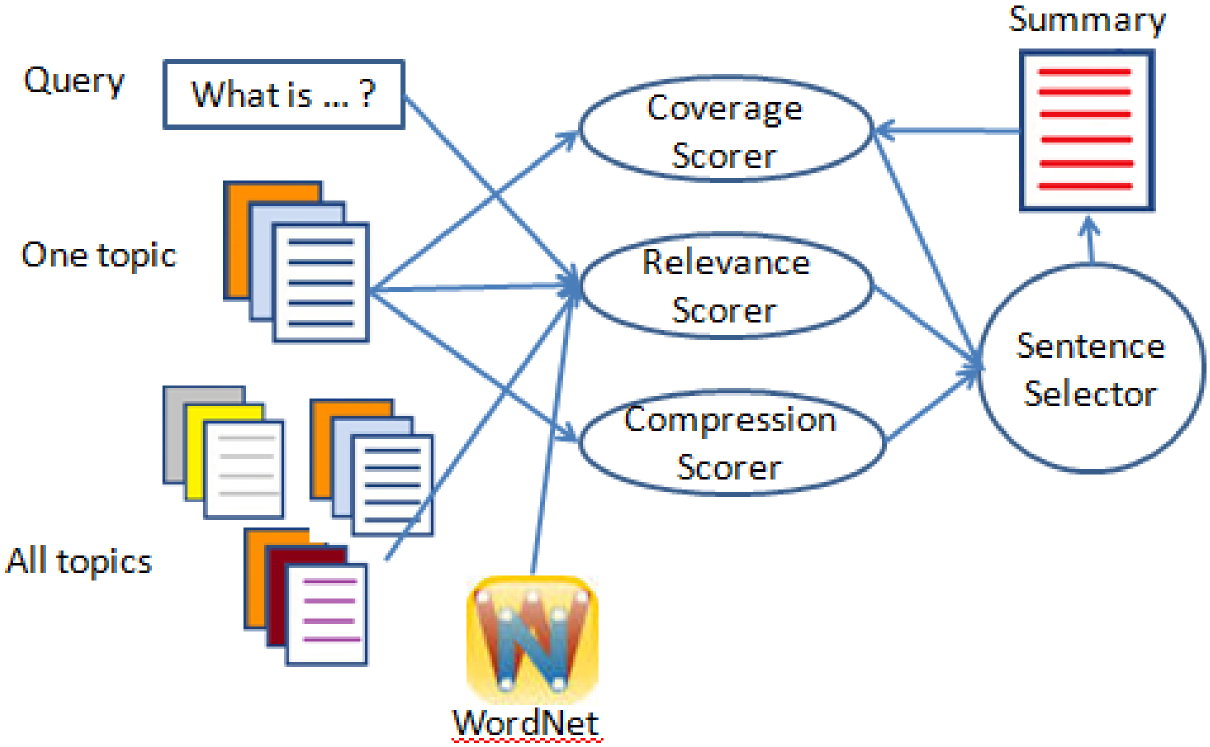


Figure 3.4: Architecture of *Comp-Rel-MCKP* document summarizer

3.3.3 Solving the problem

To solve the proposed ***Comp-Rel-MCKP*** document summarizer, I investigate the effectiveness of different algorithms including a greedy algorithm (Filatova and Hatzivassiloglou, 2004), a greedy algorithm with a performance guarantee (Takamura and Okumura, 2009), and a modified greedy algorithm for monotone and submodular function (Lin and Bilmes, 2010).

Greedy algorithm

Filatova and Hatzivassiloglou (2004) used a greedy algorithm for document summarization, which is shown in Algorithm1. In this algorithm, $f(s_l)$ denotes the score of sentence s_l which is calculated by the three measures of Coverage, Relevance, and Compression discussed in section 3.3.2. The algorithm proceeds by selecting sentence s_i with the greatest score in each iteration until it reaches the summary length.

Algorithm 1 Greedy algorithm

```
$U \leftarrow D, S \leftarrow \varnothing$
while ($U \neq \varnothing$) do
    $s_i \leftarrow \arg\, max_{s_l \in U} f(s_l)$
    if $c_i + \sum_{s_l \in S} c_l \leq K$ then $S \leftarrow S \cup \{s_i\}$
    $U \leftarrow U \backslash \{s_i\}$
end while
output $S$.
```

Greedy algorithm with performance guarantee

Khuller et al. (1999) introduced a greedy algorithm for maximum coverage problem. It was first used for document summarization by Takamura and Okumura (2009), which has a $\frac{1}{2}(1-\frac{1}{e})$ performance guarantee. The algorithm proceeds by selecting a sentence having the greatest ratio of score to its cost until it reaches the summary length. After the sequential selection to generate the summary S, its score will be compared with the highest score of all sentences and the largest will be the output.

Algorithm 2 Greedy algorithm with performance guarantee

```
$U \leftarrow D, S \leftarrow \varnothing$
while ($U \neq \varnothing$) do
    $s_i \leftarrow \arg\, max_{s_l \in U} \frac{f(s_l)}{c_l}$
    if $c_i + \sum_{s_l \in S} c_l \leq K$ then $S \leftarrow S \cup \{s_i\}$
    $U \leftarrow U \backslash \{s_i\}$
end while
$s_{max} \leftarrow \arg\, max_{s_l} f(s_l)$
if $f(S) \geq f(s_{max})$, output $S$,
otherwise, output $\{s_{max}\}$
```

Greedy algorithm for monotone and submodular function

This greedy algorithm which is illustrated in Algorithm 3 is based on the greedy algorithm proposed by Khuller et al. (1999) and has been introduced by Lin and Bilmes (2010) for the document summarization problem while having a submodular score function. Lin and Bilmes proved theoretically and empirically that their modified greedy algorithm solves the budgeted submodular maximization problem near-optimally. It has a constant factor approximation of $(1-\frac{1}{e}) \simeq 0.632$ for the cardinality constrained version of the problem and $(1-\frac{1}{\sqrt{e}})$ when using a scaled cost in the problem. It shows that the worst case bound, however, the quality of the generated summary in most cases will be much better than this bound.

Algorithm 3 Greedy algorithm for monotone and submodular function

```
$U \leftarrow D, S \leftarrow \varnothing$
**while** $(U \neq \varnothing)$ **do**
    $s_i \leftarrow \arg max_{s_l \in U} \frac{f(S\cup\{s_l\})-f(S)}{(c_l)^r}$
    **if** $c_i + \sum_{s_l \in S} c_l \leq K$ **and** $f(S \cup \{s_i\}) - f(s) \geq 0$ **then** $S \leftarrow S \cup \{s_i\}$
    $U \leftarrow U \backslash \{s_i\}$
**end while**
$s_{max} \leftarrow \arg max_{s_l} f(s_l)$
**if** $f(S) \geq f(s_{max})$, output $S$,
**otherwise,** output $\{s_{max}\}$
```

Similar to both Algorithm 1 and 2, it is based on sequential selection. In each step, it selects sentence s_i with greatest ratio of score gain to scaled cost. In the algorithm, $r \geq 0$ is a scaling factor to adjust the scale of the cost which results in having a $(1-1/\sqrt{e})$ approximation factor (see (Lin and Bilmes, 2010) for more details).

To get a near optimal solution using Algorithm 3, the scoring function $F(S)$ should be monotone and submodular[10] (Lin and Bilmes, 2011a). Otherwise, this greedy algorithm cannot guarantee a near optimal summary. In the next section, I show that the proposed objective function which is discussed in Section 3.3.2 is monotone and submodular.

Coverage function

Since penalizing redundancy violates the monotonicity property (Lin and Bilmes, 2010), I reward coverage instead, which implicitly has redundancy in its definition.

$Cov(S)$ can be interpreted as a function representing the coverage level of document set D by the summary S. The function $Cov(S)$ penalizes redundancy implicitly as redundant sentences cover fewer words and rewards coverage by selecting sentences with the greatest number of uncovered words. As soon as a sentence s_i is chosen to be in the summary S, all

[10] See section 2.2.3.

of the words forming the sentence s_i, will be ignored in calculating the coverage level of other sentences if they include the same word.

The function $Cov(S)$ has the monotonicity property as coverage is improved by adding some sentences. It also has the submodularity property. Consider two summary sets $S(A)$ and $S(B)$, where $S(B) \subseteq S(A)$. Adding a new sentence s_i to $S(B)$ increases the value of the function $Cov(S)$ more than the increment resulting from adding s_i to $S(A)$. This is because the conceptual units (words) forming the new sentence might have already been covered by those sentences that are in the larger summary $S(A)$ but not in the smaller summary $S(B)$.

Relevance function

$Rel(S)$ combines a query-related function ($sim(s_i, q)$) and an importance-oriented one ($imp(s_i)$). Both the query-related and importance-oriented functions are monotone as the similarity of summary S to the given query q is not improved by adding a sentence to it. This is because the selected sentence s_i is totally dissimilar to q and hence there is no added value for the query-related part in the worst case. In addition, the value of $imp(s_i)$, even for last sentences in a document, would result an increment in the importance-based value of a summary. It also has the submodularity property. Consider two summary sets $S(A)$ and $S(B)$, where $S(B) \subseteq S(A)$. Adding a new sentence s_i to $S(B)$ increases the value of the both functions equal to the increment resulting of adding s_i to $S(A)$ because the same sentence is added to both summaries $S(A)$ and $S(B)$ which results in the same increase in both $sim(s_i, q)$ and $imp(s_i)$.

Theorem 3.1. *Given functions $F : 2^V \rightarrow \mathbb{R}$ and $f : \mathbb{R} \rightarrow \mathbb{R}$, the composition $F' = f \circ F : 2^V \rightarrow \mathbb{R}$ (i.e., $F'(S) = f(F(S))$) is nondecreasing submodular, if f is non-decreasing concave and F is nondecreasing submodular.*

Submodular functions have some similar properties to convex and concave functions (Lovász, 1983) such as their closure under some operations including mixtures, and truncation. So, using Theorem 3.1 (Lin and Bilmes, 2011a) and the property that summation preserves submodularity, it is easy to see that $Rel(S)$ is submodular.

Compression function

The $Comp(S)$ function which is considered as the number of deleted words in the original sentences of a summary is monotone as the compression level of the summary is not worsen by adding a sentence. This is because some words might be deleted in the new sentence. It also has the submodularity property because the same sentence is added to both summary sets $S(A)$ and $S(B)$, where $S(B) \subseteq S(A)$. So, the incremental value of $Comp(S)$ by adding the same new sentence is the same for both summaries.

3.4 Summary

In this chapter, I explained the *Comp-Rel-MCKP* model for query-based extractive document summarization in detail. As its name implies, I consider three measures of *Coverage*, *Relevance*, and *Compression* jointly to score sentences in the proposed model which are calculated using unsupervised methods. Coverage considers the number of unique words in a sentence, Relevance considers the semantic similarity between a sentence and the given query and also importance of its words, and Compression considers the number of insignificant words in a sentence when scoring sentences. I discussed three greedy algorithms to select a combination of sentences to form a summary. The best performing algorithm among them has $(1 - \frac{1}{\sqrt{e}})$ performance guarantee when its scoring function is monotone and submodular. I also explained how the scoring functions for *Coverage*, *Relevance*, and *Compression* are monotone and submodular. This chapter presents the first attempt to model document summarization as a MCKP problem with the three measures of Coverage, Relevance, and Compression. It also presents how to define a Compression measure as a submoular function which enables us to integrate compression in a good performance greedy algorithm. In the next chapter, I show how I evaluate the results of the proposed approach on the DUC 2007 dataset using the ROUGE measure.

Chapter 4

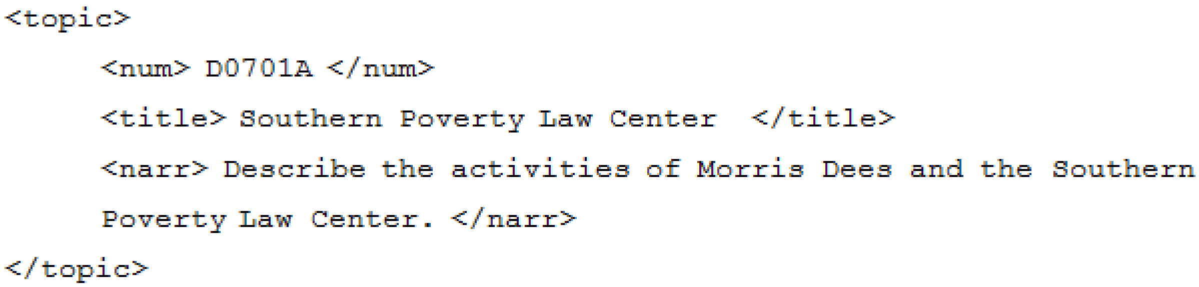
Experimental Results

4.1 Introduction

I presented the proposed multi-document summarization method in the previous chapter. It considers three measures of Coverage, Relevance, and Compression to rank sentences, in addition it applies a greedy algorithm to generate a summary. In this chapter, I present the experimental results of the proposed method and compare it with three different methods of the-state-of-the-art. The following sections explain the dataset, the summarization approaches I implemented to compare the proposed summarization method with and the experimental results.

4.2 Task Overview

The Comp-Rel-MCKP summarizer's task is to generate a summary with a length of 250 words by selecting important sentences in a collection of relevant documents with regards to a given query for a topic. Each topic has a title and a narrative which is considered as a query in query-based document summarization. A sample topic is shown below:



```
<topic>
     <num> D0701A </num>
     <title> Southern Poverty Law Center  </title>
     <narr> Describe the activities of Morris Dees and the Southern
     Poverty Law Center. </narr>
</topic>
```

4.3 Dataset

I use the data from Document Understanding Conference (DUC[11]), which is one of the main benchmarks for the document summarization field. I focus on the DUC 2007 dataset which is the latest dataset for query-based summarization. It contains 45 different topics, each with 25 relevant documents. The dataset also has multiple human written summaries as “reference summaries”, which are used to evaluate the system-generated summaries.

4.4 Evaluation

I use the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) package[12] to evaluate the results automatically. ROUGE is a well-know package for comparing the system generated summaries to a set of reference summaries written by humans. Lin (Lin, 2004) introduced four different ROUGE metrics, including ROUGE-N which considers *n*-gram co-occurrence statistics, ROUGE-L which considers the longest common subsequence, ROUGE-W which considers the weighted longest common subsequence, and ROUGE-S which considers skip-bigram co-occurrence statistics. ROUGE-N is widely used in multi-document summarization research. Also Lin and Bilmes (2011a) show that ROUGE-N is monotone and submodular. Thus, I use the ROUGE-N measure for the evaluation since the proposed method is also submodular. In the following, I explain ROUGE-N in more details.

ROUGE-N considers an *n*-gram overlap between a system-generated summary and a set of human generated summaries\reference summaries. It is defined as follows:

$$ROUGE-N = \frac{\sum_{S\in\{ReferenceSummaries\} gram_n\in S}\sum Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries\} gram_n\in S}\sum Count(gram_n)} \tag{4.1}$$

where n indicates the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ shows the maximum number of n-grams co-occurring in a system generated summary and a set of reference

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

[12] ROUGE package is available at http://www.berouge.com

summaries, respectively. ROUGE-N is a recall-related measure since the denominator of Equation 4.1 is the total sum of the number of n-grams occurring at the reference summary side.

In this thesis, I focus on ROUGE-1 (unigram) and ROUGE-2 (bigram) scores, and report precision, recall and F-measure for evaluation since these metrics are found to correlate well with human judgment and widely used to evaluate an automatic summarizer (McDonald, 2007; Lin and Bilmes, 2011a; Dasgupta et al., 2013). I adopt the definition of recall, precision and F-measure from (Hasan, 2013). Recall for document summarization is interpreted is the ratio of the number of common words of the system generated and the human generated summaries to the total number of words in the human generated summary. Precision as the ratio of the number of common words of the system generated and the human generated summaries to the total number of words in the system generated summaries. F-measure is a combination of precision and recall to evaluate the overall performance.

4.5 Comparison with the State-Of-The-Art

4.5.1 Baseline

I adopt the baseline from DUC 2007[13]. It concatenates leading sentences of all relevant documents up to the length limit.

4.5.2 Rel-MCKP

In this method, a summary is generated using the summarization method proposed by Takamura and Okumura (2009). They consider MCKP to model summarization and consider *Relevance* and *Coverage* measures in the sentence selection process. Two different greedy algorithms introduced in Section 3.3.3 proposed by (Filatova and Hatzivassiloglou, 2004) and (Takamura and Okumura, 2009) are applied to generate a summary. These two systems are shown by *Rel-MCKP-Greedy* and *Rel-MCKP-Greedy-Per* respectively in the comparisons.

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

4.5.3 Comp-Rel-MCKP

In this method, a summary is generated using the proposed Comp-Rel-MCKP method in which all scoring functions are submodular and monotone and the three measures of *Coverage*, *Relevance*, and *Compression* are considered. A modified greedy algorithm introduced in (Lin and Bilmes, 2010) for submodular functions which has a performance guarantee of $(1-\frac{1}{\sqrt{e}})$ is used to generate the summary. This method is referred to as *Comp-Rel-MCKP* in the comparisons.

4.6 Experiments

In this section, I present the experimental results of the proposed method and compare it with the methods discussed in the previous section. The main goal of the experiments is to show the effectiveness and efficiency of each method. I conduct a series of experiments on the DUC dataset introduced in Section 4.3. In the experiments, I first investigate the effects of different parameters on the performance of the proposed method. Then, I compare the proposed method with some previous summarization methods.

4.6.1 Experiment 1

In this experiment, I investigate the effect of the cost scaling factor, r, which is used in Algorithm 3 to adjust the scale of the cost. The result of the experiment for different cost scaling factors are shown in Figure 4.1 and 4.2, based on ROUGE-1 and ROUGE-2. The scaling factor ranges from 0.8 to 2. As the diagram shows the scaling factor to 1.2, $r = 1.2$ results in better performance with respect to recall, precision, and F-measure.

4.6.2 Experiment 2

In this experiment, I investigate the effect of employing the stemming algorithm which is used in preprocessing step on the relevance of a sentence and a query which was discussed in Section 3.3.2. Stemming was helpful in the proposed method, specially for finding the similarity of a pair of words containing plural nouns. Stemming plural nouns allowed

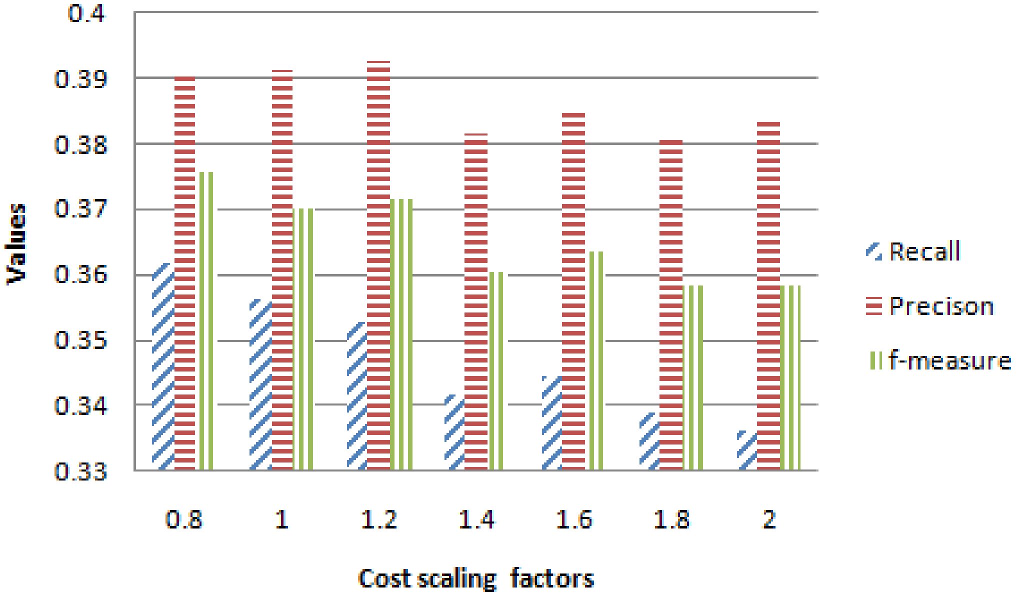


Figure 4.1: Values of Recall, Precision, and F-measure of ROUGE-1 for different scaling factors

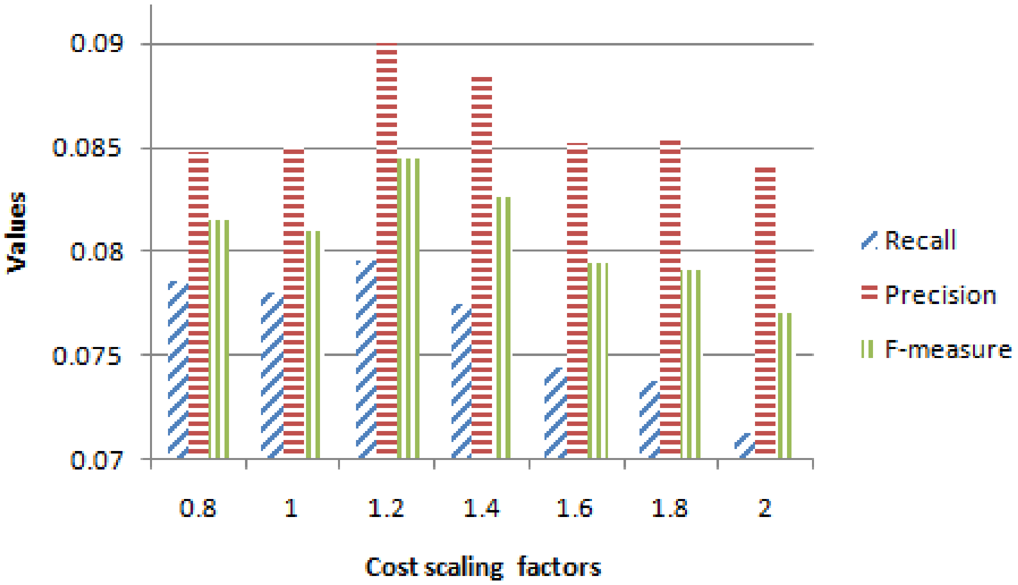


Figure 4.2: Values of Recall, Precision, and F-measure of ROUGE-2 for different scaling factors

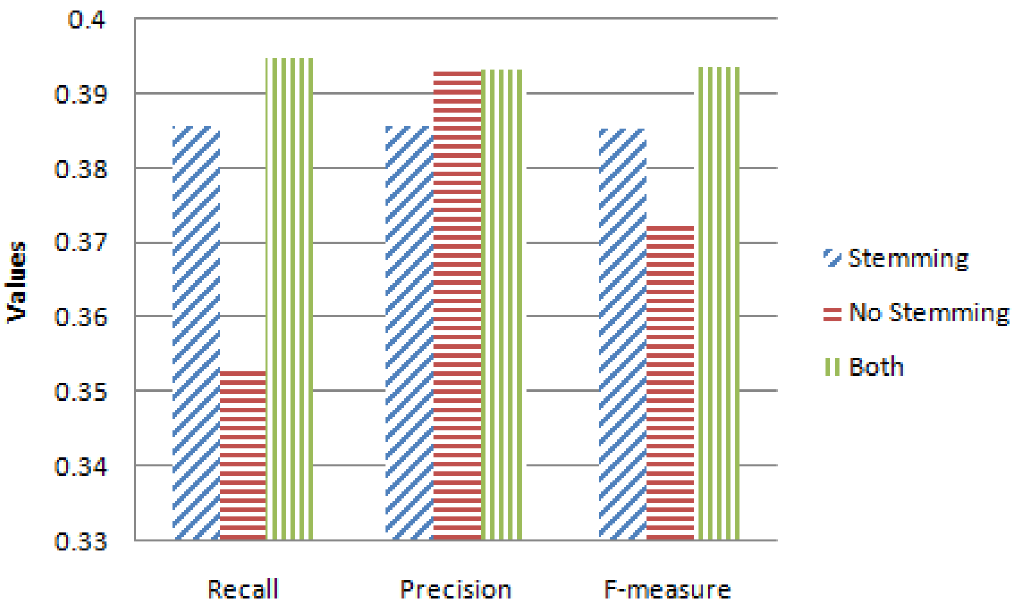


Figure 4.3: Values of Recall, Precision, and F-measure of ROUGE-1 for different stemming strategies

us find the words easily in WordNet, while non-stemmed words in plural form cannot be found in WordNet. For example, words such as “schools” or “programs” cannot be found in WordNet. So, the similarity measures consider no correlation between them as they are not in WordNet. On the other side, stemming algorithms such as Porter (Porter, 1980) do not find the stems of many words correctly. As an illustration, these algorithms eliminate “e” which exists at the end of most words such as “article” or “revoke”, and result in a word which does not have any corresponding concept in WordNet, while looking up most non-stemmed words in WordNet such as “articles” leads to a matching concept in WordNet. Therefore, I run an experiment in which I consider each word in both stemmed and original form in the process of calculating the similarity between pairs of words of a sentence and a query to have the advantageous of both stemming and not stemming. Figure 4.3 and 4.4 illustrate the result of this experiment in form of ROUGE-1 and ROUGE-2.

I consider three cases, including 1) *Stemming* in which I apply the Porter stemmer (Porter, 1980) and consider the stemmed word to measure the similarity between a sentence and a query, 2) *No Stemming* in which I consider words in the original form, and 3)

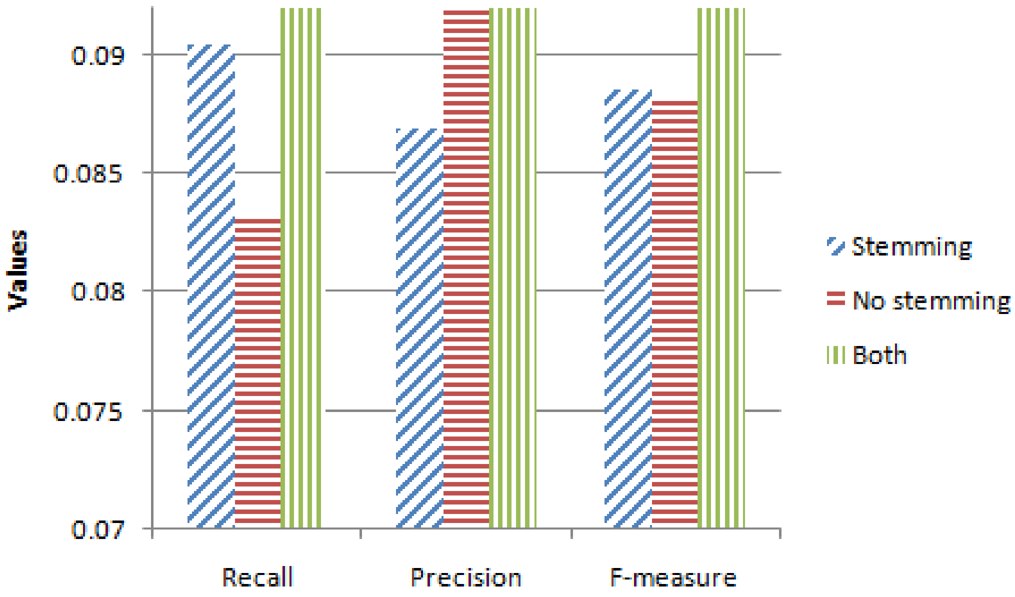


Figure 4.4: Values of Recall, Precision, and F-measure of ROUGE-2 for different stemming strategies

Both in which I consider both stemmed and not-stemmed words in similarity measurement and consider the higher one as the similarity between them. In this experiment, the value of the scaling factor is 1.2 for all three cases.

I find that considering both stemmed and not-stemmed words results in having a better performance with respect to all three measures of Recall, Precision, and F-measure. In spite of the above mentioned mistakes in stemming words, the performance of the proposed summarizer outperforms with respect to Recall, and F-measure in the case of using stemmed words compared to the case of using the original form of the words. Considering both stemmed and not-stemmed words increases the complexity of the calculation and makes the system slow. Therefore, I consider just stemmed words to calculate the relevance measure in the similarity measurement for future experiments.

4.6.3 Experiment 3

In this experiment, I investigate the effect of using the title or both the title and narrative (query), which was introduced in Section 4.2, in measuring the relevance of a sentence to the topic. In this experiment, the value of the scaling factor is 1.2 for all three cases. The value of Precision, Recall, and F-measure for ROUGE-1 and ROUGE-2 are illustrated in Figure 4.5 and 4.6. It is evident that using both title and narrative to improve the value of all three measures of Recall, Precision, and F-measure.

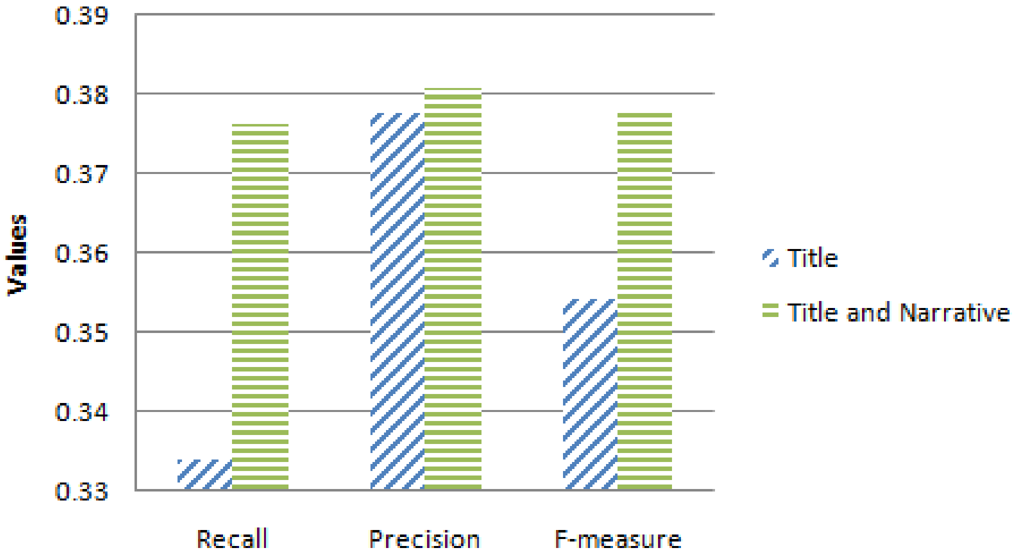


Figure 4.5: Values of Recall, Precision, and F-measure of ROUGE-1 considering title or both title and narrative

4.6.4 Experiment 4

In this experiment, I investigate the effect of using the semantic similarity measure which was introduced in Section 3.3.2 in measuring the relevance of a sentence to the topic. The values of Precision, Recall, and F-measure for ROUGE-1 and ROUGE-2 are illustrated in Figure 4.7 and 4.8.

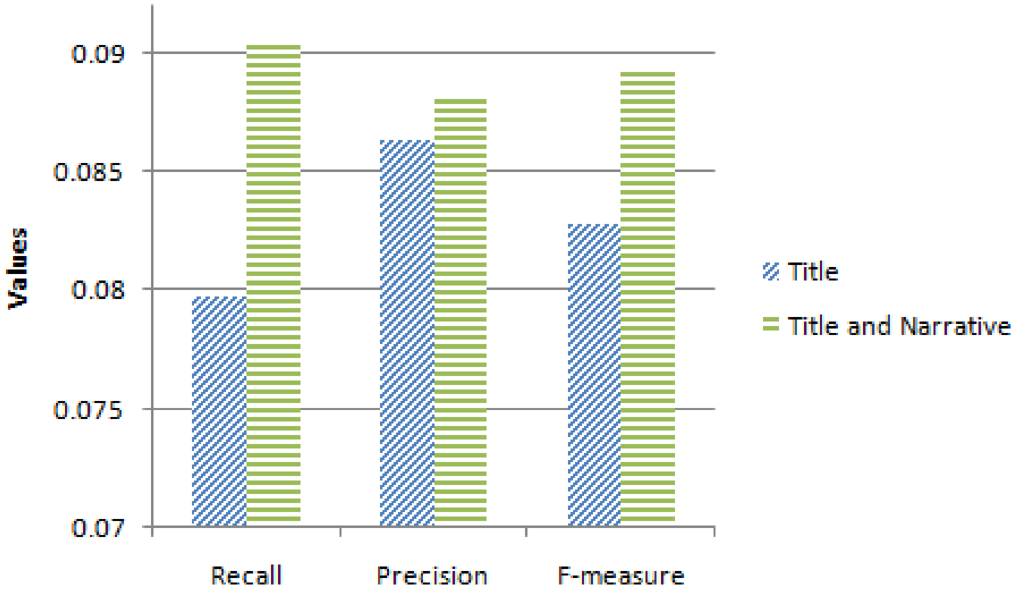


Figure 4.6: Values of Recall, Precision, and F-measure of ROUGE-2 considering title or both title and narrative

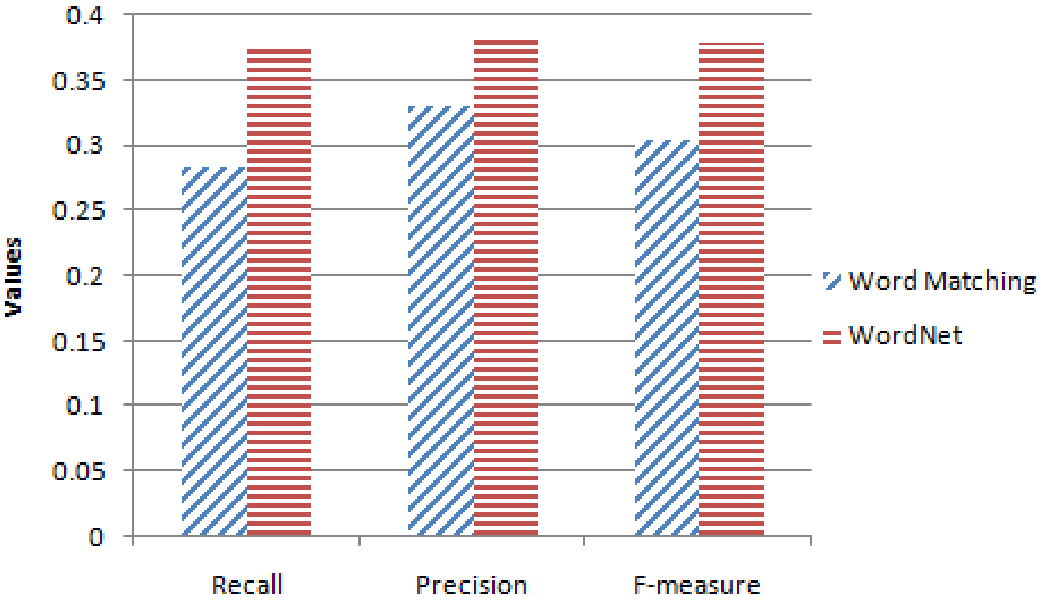


Figure 4.7: Values of Recall, Precision, and F-measure of ROUGE-1 for WordNet and Word Matching based similarity measures to calculate Relevance

As I predicted, using WordNet based measures improve the value of all three measures of Recall, Precision, and F-measure compared to the word-matching based measure.

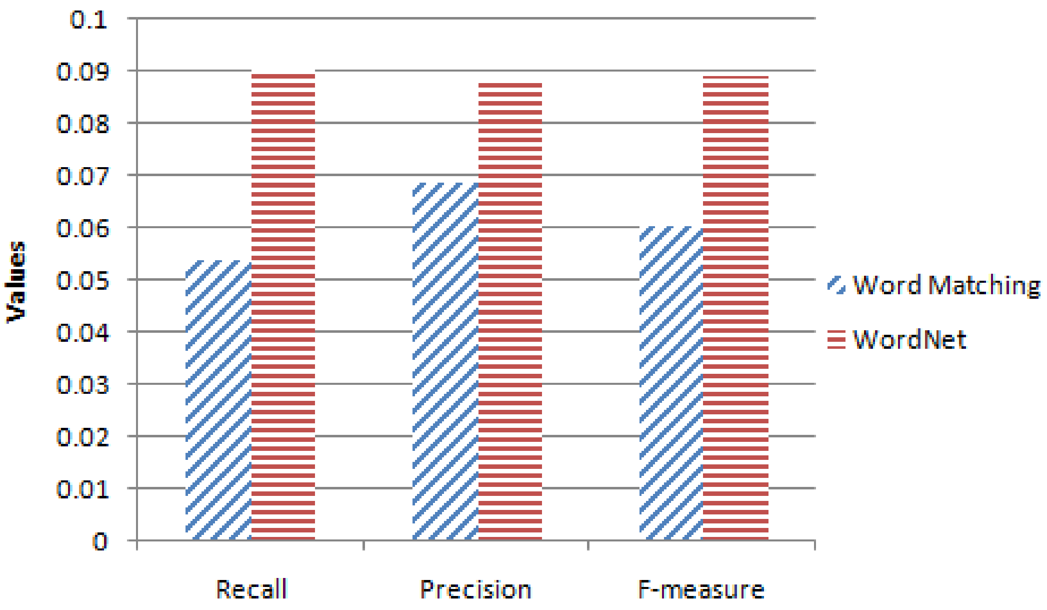


Figure 4.8: Values of Recall, Precision, and F-measure of ROUGE-2 for WordNet and Word Matching based similarity measures to calculate Relevance

4.6.5 Experiment 5

In this experiment, I investigate the performance of different summarization approaches which were introduced in Section 4.5. The result is shown in Table 4.1 which compares the values of ROUGE-1 and ROUGE-2 measures of the approaches. Precision, Recall, and F-measure metrics are abbriviated to P, R, and F respectively in this table. The best scores are bolded for each measure. As it is illustrated, the *Comp-Rel-MCKP* and *Rel-MCKP-Greedy-Per* approaches outperform the two other approaches for all measures. The proposed approach, *Comp-Rel-MCKP*, has a better performance for most measures compared to *Rel-MCKP-Greedy-Per* approach.

Table 4.1: ROUGE-1 and ROUGE-2 evaluation of different approaches on the DUC 2007 Dataset

Metrics		**Methods**			
		Baseline	Rel-MCKP-Greedy	Rel-MCKP-Greedy-Per	Comp-Rel-MCKP
Rouge-1	P	0.3737	0.3769	0.3808	**0.3809**
	R	0.3334	0.3318	0.3661	**0.3763**
	F	0.3522	0.3527	0.3731	**0.3782**
Rouge-2	P	0.0654	0.0827	**0.0911**	0.08821
	R	0.0638	0.0739	0.0782	**0.0904**
	F	0.0644	0.0780	0.0837	**0.0892**

Figure 4.9 and 4.10 also illustrate the result which provides a better view of the performance. As it is illustrated, Comp-Rel-MCKP and Rel-MCKP-Greedy-Per approaches outperform the two other approaches for all three metric of recall, precision and f-measure. And also the proposed approach (Comp-Rel-MCKP), has a better performance for most measures compared Rel-MCKP-Greedy-Per approach. The results demonstrate that our Comp-Rel-MCKP summarizer which combine three submodular measures of compression, coverage, and relevance achieves better performance compared to the other summarization systems that use two non-submodular measures of relevance and coverage.

The results demonstrate that the Comp-Rel-MCKP summarizer and , which combine three submodular measures of compression, coverage, and relevance achieves better performance compared to the other summarization systems that use two non-submodular measures of relevance and coverage.

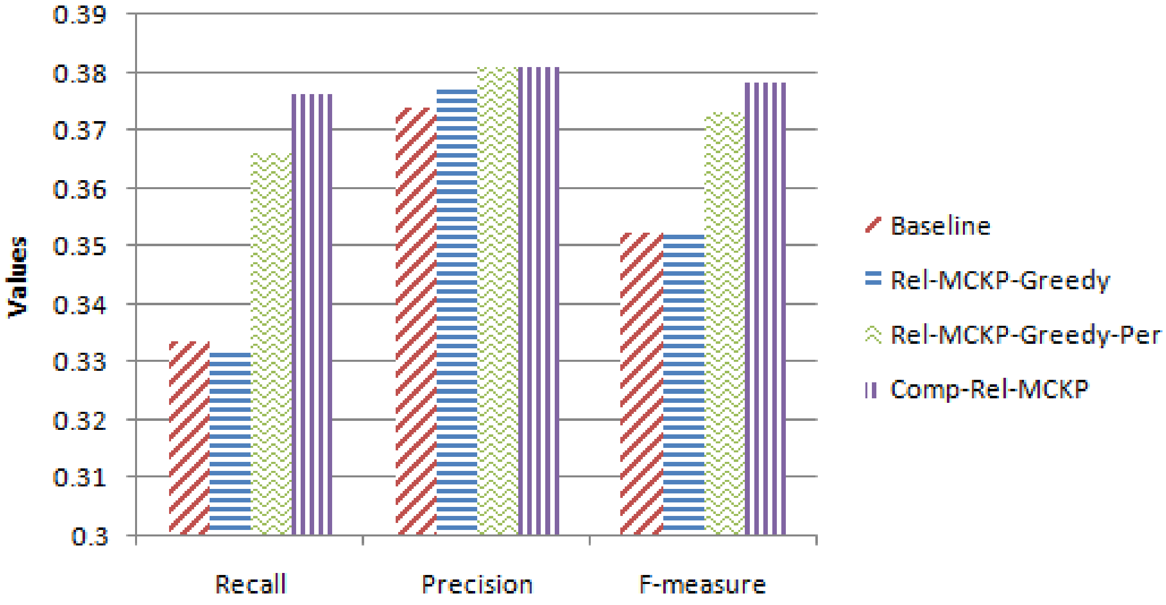


Figure 4.9: Values of Recall, Precision, and F-measure of ROUGE-1 of different approaches

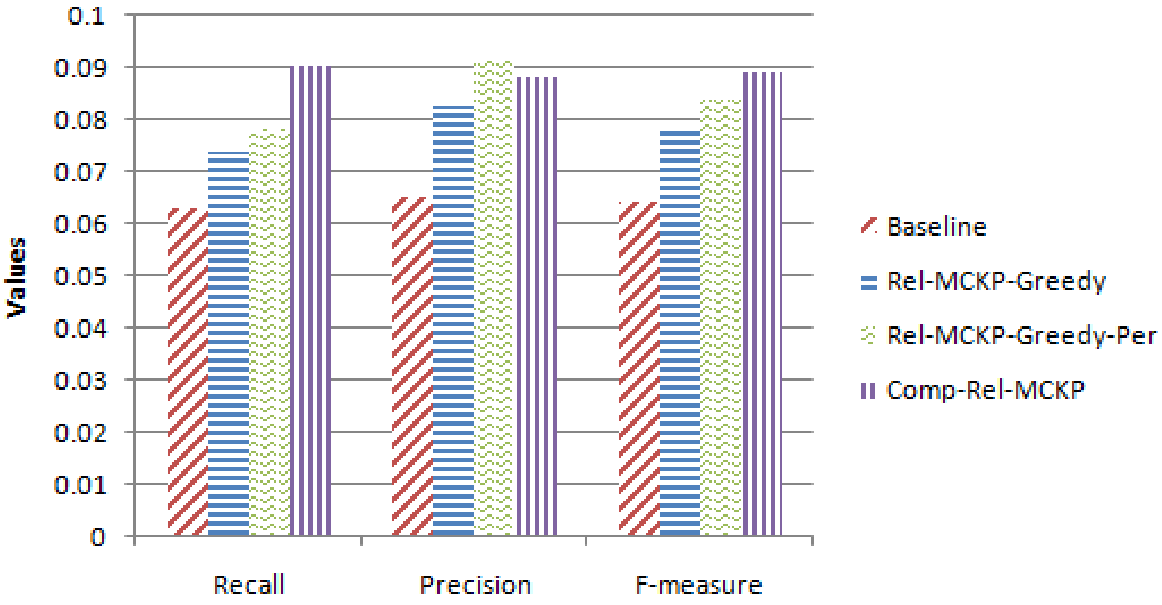


Figure 4.10: Values of Recall, Precision, and F-measure of ROUGE-2 of different approaches

4.7 Summary

In this chapter, the evaluation procedure and results were discussed. First, I explained the task of automatic document summarizer, DUC data sets and ROUGE measures which evaluate the system generated summaries with regards to human generated summaries. Then, I investigated how different cost scaling factors affect the performance of the proposed summarizer in terms of Recall, Precision, and F-measure. In addition, I ran different experiments to investigate whether or not to use stemming, or whether or not to consider both title and narrative provided within each topic. Finally, I compared the results of the Comp-Rel-MCKP summarizer with three methods introduced in literatures. As illustrated, the proposed method provides improvements over the existing approaches.

Chapter 5

Conclusion and Future Works

5.1 Introduction

This thesis studied the problem of multi-document summarization. In this last chapter, I conclude by summarizing the proposed method and contributions made towards improving the performance of existing document summarization models. I also suggest some future directions that can enhance the efficiency of the proposed model.

5.2 Thesis Summary

The problem of document summarization has been studied extensively since the 1950s because of its key role in reducing the volume of information a user has to read, and consequently the amount of time which is required to read the relevant documents to find the desired information. To this aim, numerous document summarization models have been developed. Document summarization models broadly fall into two categories of *Extractive* and *Abstractive* considering the strategy of generating a summary. In extractive document summarization, the sentences from the documents are extracted to form a summary, while in abstractive document summarization, important information of the documents is rewritten as a summary. On the other hand, in most cases users desire to find information about a specific topic which results in another categorization of *Query-based* and *Generic* summarization.

In the document summarization domain, improving existing summarization models and the quality of generated summaries are always essential. So, my main goal in this thesis is to improve the quality of the system-generated summaries. In this thesis, I present an automatic document summarizer which generates a summary for multiple relevant documents. I proposed an extractive document summarization model and my focus was on query based summarization. However, the proposed model can be generalized and also applied for generic document summarization. I modeled the extractive document summarization problem based on the Maximum Coverage with KnaPsack constraint (MCKP) problem. The main motivation behind mapping the document summarization problem to a MCKP problem was its great performance and natural fit for the summarization domain. The effectiveness of extractive document summarizers deeply relies on how I identify the importance of sentences. I use three metrics of Coverage, Relevance, and Compression to estimate the scores of sentences. The coverage metric assign scores to sentences based on how they represent the document. The Relevance metric considers the importance level of information in a sentence in addition to the similarity level of the sentence and the given query. The compression metric considers the number of deletable words of a sentence. The proposed model, which is called *Comp-Rel-MCKP*, is an improvement of previous MCKP based models in which I considers compression as an extra metric to decide on the importance of a sentence.

Query based summarization deeply relies on methods to identify the relevance of sentences to the given query. A key aspect of the proposed approach to calculate the Relevance score was the use of WordNet to discover the correlation between a sentence and a query semantically. The reason to use WordNet based measures is their efficiency and effectiveness in determining the semantic correlation between words and their advantageous over word matching or co-occurrence based measures.

As I discussed earlier, I use three measures of Coverage, Relevance, and Compression to assign a score to each sentence of the relevant documents. Then, I apply a modified greedy algorithm which has a performance guarantee of $(1 - \frac{1}{\sqrt{e}})$ to generate a summary. The scoring function should be monotone and submodular to guarantee the performance which I considered in the definition of the scoring functions.

I evaluated the proposed summarization model on the DUC 2007 dataset which is for a query-based summarization task. I investigated the effect of different parameters of the proposed model such as using WordNet and stemming. The experiments and evaluations illustrated that stemming words and using semantic similarity measures to calculate the relevance of a sentence and a query increase the quality of the summaries. I compared the Comp-Rel-MCKP summarization system with a baseline, and two recent MCKP based summarization models. The results on the DUC 2007 data sets showed the effectiveness of the proposed approach.

5.3 Future Works

Some extensions to the proposed model are summarized as follows:

- A better estimation of the relevance of a sentence to a query plays a key role in quality of query based document summarization. In spite of the good performance of WordNet based similarity measures, there might be a case in which a word is not in WordNet due to the ever increasing number of new words. So, I plan to apply search engine based and co-occurrence similarity measures beside WordNet based measures to calculate the relevance of a sentence and a query. Search engine based similarity measures usually use web page counts of words w_a and w_b, and combinations of them together and also snippets retrieved from a web search engine to calculate the similarity of word w_a and word w_b. Co-occurrence based measures estimate the similarity between any pairs of words w_a and w_b by considering frequencies of co-occurring words w_a and w_b in a document together.

- The proposed document summarizer does not detect the references of pronouns and does not replace them with their corresponding names. This problem which is called anaphora problem can really affect the readability and coherence of the generated summary. Two different kinds of approaches have been introduced for anaphora problem, including knowledge-rich and knowledge-poor approaches. Applying these approaches can improve the quality of the system generated summaries.

Bibliography

P. B. Baxendale. Machine-made index for technical literature: an experiment. *IBM Journal of Research and Development*, 2(4):354–361, 1958.

T. Berg-Kirkpatrick, D. Gillick, and D. 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.

J. Carbonell and J. 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.

A. Celikyilmaz and D. 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.

Y. Chali and S. 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 2012)*, pages 457–474. Citeseer, 2012.

J. Clarke and M. Lapata. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, pages 399–429, 2008.

H. T. Dang. Overview of duc 2005. In *Proceedings of the Document Understanding Conference*, 2005.

D. Das and A. F. Martins. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, 4:192–195, 2007.

A. Dasgupta, R. Kumar, and S. Ravi. Summarization through submodularity and dispersion. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1014–1022, 2013.

H. C. Daume. *Practical structured learning techniques for natural language processing*. PhD thesis, University of Southern California, 2006.

H. P. Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16 (2):264–285, 1969.

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

E. Filatova and V. Hatzivassiloglou. A formal model for information selection in multi-sentence text extraction. In *Proceedings of the 20th international conference on Computational Linguistics*, page 397. Association for Computational Linguistics, 2004.

M. Galley. A skip-chain conditional random field for ranking meeting utterances by importance. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 364–372. Association for Computational Linguistics, 2006.

M. Galley and K. McKeown. Lexicalized markov grammars for sentence compression. In *Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 180–187, 2007.

P.-E. Genest and G. Lapalme. Fully abstractive approach to guided summarization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 354–358. Association for Computational Linguistics, 2012.

D. Gillick and B. Favre. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing of North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL HLT)*, pages 10–18, 2009.

D. Gillick, B. Favre, and D. Hakkani-Tur. The icsi summarization system at tac 2008. In *Proceedings of the Text Understanding Conference*, 2008.

D. Gillick, B. Favre, D. Hakkani-Tur, B. Bohnet, Y. Liu, and S. Xie. The icsi/utd summarization system at tac 2009. In *Proc. of the Text Analysis Conference, Gaithersburg, MD (USA)*, 2009.

J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization-Volume 4*, pages 40–48. Association for Computational Linguistics, 2000.

G. Grefenstette. Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In *Working notes of the AAAI Spring Symposium on Intelligent Text summarization*, pages 111–118, 1998.

S. Hasan. *Complex question answering: minimizing the gaps and beyond*. PhD thesis, Lethbridge, Alta.: University of Lethbridge, Dept. of Mathematics and Computer Science, 2013.

E. Hovy, C.-Y. Lin, L. Zhou, and J. Fukumoto. Automated summarization evaluation with basic elements. In *Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC 2006)*, pages 604–611. Citeseer, 2006.

H. Jing. Sentence reduction for automatic text summarization. In *Proceedings of the sixth conference on Applied natural language processing*, pages 310–315. Association for Computational Linguistics, 2000.

S. Khuller, A. Moss, and J. S. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

K. Knight and D. Marcu. Summarization beyond sentence extraction: A probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107, 2002.

J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73. ACM, 1995.

C. Li, F. Liu, F. Weng, and Y. Liu. Document summarization via guided sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 490–500, 2013.

Y. Li, Z. A. Bandar, and D. McLean. An approach for measuring semantic similarity between words using multiple information sources. *Knowledge and Data Engineering, IEEE Transactions on*, 15(4):871–882, 2003.

C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of Association for Computational Linguistics*, pages 74–81, 2004.

H. Lin. *Submodularity in Natural Language Processing: Algorithms and Applications*. PhD thesis, University of Washington, 2012.

H. Lin and J. 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.

H. Lin and J. 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, 2011a.

H. Lin and J. Bilmes. Word alignment via submodular maximization over matroids. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 170–175. Association for Computational Linguistics, 2011b.

H. Lin, J. Bilmes, and S. Xie. Graph-based submodular selection for extractive summarization. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*, pages 381–386. IEEE, 2009.

L. Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

I. Mani and M. T. Maybury. *Advances in automatic text summarization*, volume 293. MIT Press, 1999.

D. Marcu. From discourse structures to text summaries. In *Proceedings of the Workshop on Intelligent Scalable Text Summarization*, volume 97, pages 82–88. Citeseer, 1997.

A. F. Martins and N. 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.

M. McCord. Slot grammar: A system for simpler construction of practical natural language grammars. In *Proceedings of the International Symposium on Natural Language and Logic*, pages 118–145, 1989.

R. McDonald. A study of global inference algorithms in multi-document summarization. In *Advances in Information Retrieval*, pages 557–564. Springer, 2007.

R. T. McDonald. Discriminative sentence compression with soft syntactic evidence. In *Proceeding of the conference of European Association for computational linguistics (EACL)*, 2006.

G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4):235–244, 1990.

H. Morita, T. Sakai, and M. 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.

H. Morita, R. Sasano, H. Takamura, and M. Okumura. Subtree extractive summarization via submodular maximization. In *Proceeding of 51st Annual Meeting of the Association for Computational Linguistics*, pages 1023–1032, 2013.

N. K. Nagwani and S. Verma. A frequent term and semantic similarity based single document text summarization algorithm. *International Journal of Computer Applications (0975–8887) Volume*, pages 36–40, 2011.

A. Nenkova and K. McKeown. A survey of text summarization techniques. In *Mining Text Data*, pages 43–76. Springer, 2012.

M. Osborne. Using maximum entropy for sentence extraction. In *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*, pages 1–8. Association for Computational Linguistics, 2002.

S. Petrov and D. Klein. Learning and inference for hierarchically split pcfgs. In *Proceedings of the national conference on artificial intelligence*, volume 22, pages 1663–1666, 2007.

G. Pirró and J. Euzenat. A feature and information theoretic framework for semantic similarity and relatedness. In *The Semantic Web–ISWC 2010*, pages 615–630. Springer, 2010.

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

R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1):17–30, 1989.

D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

B. Schiffman, A. Nenkova, and K. McKeown. Experiments in multidocument summarization. In *Proceedings of the second international conference on Human Language Technology Research*, pages 52–58. Morgan Kaufmann Publishers Inc., 2002.

N. Seco, T. Veale, and J. Hayes. An intrinsic information content metric for semantic similarity in wordnet. In *Proceeding of the European Conference on Artificial Intelligence (ECAI 2004)*, volume 16, page 1089, 2004.

S. Sekine and C. Nobata. Sentence extraction with information extraction technique. In *Proceedings of the Document Understanding Conference*, 2001.

D. Shen, J.-T. Sun, H. Li, Q. Yang, and Z. Chen. Document summarization using conditional random fields. In *Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI)*, volume 7, pages 2862–2867, 2007.

R. Sipos, P. Shivaswamy, and T. 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.

H. Takamura and M. 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.

J. Turner and E. Charniak. Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 290–297. Association for Computational Linguistics, 2005.

L. Wang, H. Raghavan, V. Castelli, R. Florian, and C. Cardie. A sentence compression based framework to query-focused multi-document summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1384–1394, 2013.

W.-t. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. Multi-document summarization by maximizing informative content-words. In *Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI)*, volume 2007, pages 1776–1782, 2007.

D. Zajic, B. J. Dorr, J. Lin, and R. Schwartz. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing & Management*, 43(6):1549–1570, 2007.

Appendix A

List of Stop Words

PRONOUNS FORMS

i	me	my	myself
we	us	our	ours
ourselves	you	your	yours
yourself	yourselves	he	him
his	himself	she	her
hers	herself	it	its
itself	they	them	their
theirs	themselves	what	which
who	whom	this	that
these	those		

VERB FORMS

BE

am	is	are	was
were	be	been	being

HAVE

have	has	had	having

DO

do	does	did	doing

AUXILIARIES

will	would	shall	should
can	could	may	might
must	ought		

COMPOUND FORMS

i’m	you’re	he’s	she’s
it’s	we’re	they’re	i’ve
you’ve	we’ve	they’ve	

i’d	you’d	he’d	she’d
we’d	they’d	i’ll	you’ll

he’ll	she’ll	we’ll	they’ll

isn’t	aren’t	wasn’t	weren’t
hasn’t	haven’t	hadn’t	doesn’t
don’t	didn’t		

won’t	wouldn’t	shan’t	shouldn’t
can’t	cannot	couldn’t	mustn’t

let’s	that’s	who’s	what’s
here’s	there’s	when’s	where’s
why’s	how’s		

ARTICLES

a	an	the	

THE REST

and	but	if	or
because	as	until	while
of	at	by	for
with	about	against	between
into	through	during	before
after	above	below	to
from	up	down	in
out	on	off	over
under			
again	further	then	once
here	there	when	where
why	how	all	any
both	each	few	more
most	other	some	such
no	nor	not	only
own	same	so	than
too	very	one	every
least	less	many	now
ever	never	say	says
said	also	get	go
goes	just	made	make
put	see	seen	whether
like	well		
back	even	still	way
take	since	another	however

two three four five
first second new old
high long

Appendix B

Sample summaries

Sample Summaries for Topic-D0723F (DUC-2007)

Topic Title
Southern Poverty Law Center

Query
Describe the activities of Morris Dees and the Southern Poverty Law Center.

Human Generated Summary

Morris Dees was co-founder of the Southern Poverty Law Center (SPLC) in 1971 and has served as its Chief Trial Counsel and Executive Director. The SPLC participates in tracking down hate groups and publicizing their activities in its Intelligence Report, teaching tolerance and bringing lawsuits against discriminatory practices and hate groups. As early as 1973 the SPLC won a federal case which forced funeral homes throughout the U.S. to provide equal services to blacks and whites. In 1991 it started a classroom program ”Teaching Tolerance” which features books, videos, posters and a magazine that goes to more than 400,000 teachers. It also funded a civil rights litigation program in Georgia to provide free legal assistance to poor people. The SPLC's most outstanding successes, however, have been in its civil lawsuits against hate groups. Dees and the SPLC have fought to break the organizations by legal action resulting in severe financial penalties. Described as ”wielding the civil lawsuit like a Buck Knife, carving financial assets out of hate group leaders,” the technique has been most impressive: 1987-$7 million against the United Klans of America in Mobile, Alabama; 1989-$1 million against Klan groups in Forsyth County, Georgia; 1990-$9 million against the White Aryan Resistance in Portland, Oregon; and 1998-$20 million against The Christian Knights of the Ku Klux Klan in Charleston, South Carolina. But despite these judgments the Ku Klux Klan and White Aryan Resistance have survived.

Baseline Summarizer

White supremacist arrested after buying hand grenades from an undercover agent said he wanted to send mail bombs to Washington and Montgomery, authorities said. The city council has declared the Ku Klux Klan a terrorist group and condemned other hate groups as well in a resolution approved after an emotional debate. A lawyer who specializes in bankrupting hate groups is going after the Aryan Nations, whose compound in the Idaho woods has served as a clubhouse for some of America's most violent racists. One of two men convicted of assaulting a woman and her son outside the headquarters of the Aryan Nations denied being a member of the white supremacist group Thursday during testimony in a civil rights case filed against them, the Aryan Nations and the group 's founder, Richard Butler. A jury on Thursday awarded $ 6.3 million to a woman and her son who were attacked by Aryan Nations guards outside the white supremacist group's north Idaho headquarters. Northern Idaho, predominantly white and rural, has been home for nearly three decades to the racist Aryan Nations. Aryan Nations leader Richard Butler vowed Saturday Richard Butler will not leave northern Idaho, despite a $ 6.3 million judgment against

Richard Butler's racist organization. Facing eviction from its compound in northern Idaho, the Aryan Nations may move its annual white supremacist gathering to Pennsylvania next year.

Rel-MCKP Summarizer Using Greedy Algorithm with Performance Guarantee

Aryan Nations leader Richard Butler declined to talk with reporters Friday. Dees has publicly said Keenans hopes to bankrupt the Aryan Nations. Race is a religion with Butler. Richard Girnt Butler hopes the rest of the world is too. Dees sought to destroy the corporate Klan. And Richard Girnt Butler believes the lawsuit is part of that war. Inside the compound, Shane Wright, an Aryan Nations security guard, thought Ku Klux Klan heard a gunshot. Steele said Butler preaches nonviolence, except in self-defense. The Klan is gone. Dees has always gotten props for dropping dimes on the hateful activities of such groups as the Ku Klux Klan and Aryan Nation. Richard Butler said Richard Butler may seek a new trial. The suit contends that not only the guards but also the Aryan Nations, its 82-year-old founder, Richard Girnt Butler, and Dees Jr. Mouzon plans a church picnic for Ku Klux Klan's country community . People who study hate groups consider Butler a godfather of the white supremacist movement. While Butler has never been convicted of any crime, Ku Klux Klan's group has attracted neo-Nazis, supremacists and separatists to the region. Edgar Steele, who represents Butler and the Aryan Nations, said he will move for a new trial.

Comp-Rel-MCKP Summarizer Using Modified Greedy Algorithm

Dees has publicly said Keenans hopes to bankrupt the Aryan Nations. Dees sought to destroy the corporate Klan. Dees has always gotten props for dropping dimes on the hateful activities of such groups as the Ku Klux Klan and Aryan Nation. The suit contends that not only the guards but also the Aryan Nations, its 82-year-old founder, Richard Girnt Butler, and Dees Jr.Morris S. Dees, of the Montgomery, Ala.But putting a hate group out of business isn't easy: While Dees has won significant civil judgments against the Ku Klux Klan and the White Aryan Resistance, the groups have survived. Hate groups are paying attention to Dees' tactics. Dees Jr. went to court in Coeur d'Alene, Idaho. The Portland case is similar to the Keenan lawsuit, in that Dees argued that White Aryan Resistance founders Tom and John Metzger incited the skinheads to commit murder. Morris S.Morris S.Butler himself and 12 other white supremacist leaders were arrested in 1987 on federal sedition charges but were acquitted at trial in Dees Jr., Ark.The Keenans' attorney, Morris Dees, had asked the jury to award more than $ 11 million in punitive damages. Dees has long used lawsuits to destroy the finances of hate groups.