ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

MULTI-DOCUMENT SUMMARIZATION USING DISTORTION-RATE RATIO

M.Sc. THESIS

Ulukbek Attokurov

Department of Computer Engineering

Computer Engineering Programme

JULY 2014

ISTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

MULTI-DOCUMENT SUMMARIZATION USING DISTORTION-RATE RATIO

M.Sc. THESIS

Ulukbek Attokurov
(504101530)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Uluğ Bayazıt

JULY 2014

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BOZULUM-HIZ ORANINA GÖRE ÇOKLU METIN ÖZETININ ÇIKARILMASI

YÜKSEK LİSANS TEZİ

Ulukbek Attokurov
(504101530)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Uluğ Bayazıt

TEMMUZ 2014

Ulukbek Attokurov, a M.Sc. student of ITU Institute of Science and Technology 504101530 successfully defended the thesis entitled **"MULTI-DOCUMENT SUMMARIZATION USING DISTORTION-RATE RATIO"**, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Uluğ Bayazıt** Istanbul Technical University	.....
Jury Members :	**Assoc. Prof. Banu DIRI** Yıldız Technical University	.....
	Assoc. Prof. Ahmet CÜNEYD TANTUĞ Istanbul Technical University	.....
		.....

Date of Submission : **05 May 2014**
Date of Defense : **26 May 2014**

To my parents,

FOREWORD

This thesis investigates multi-document summarization in the context of distortion-rate framework. Text summarization is considered as a data compression task. Hierarchical Agglomerative Clustering and optimal tree pruning algorithms are incorporated in order to detect and to eliminate the redundancy.

This thesis was completed in two years. A paper that outlines the main concepts of the thesis was published in ACL(Association for Computational Linguistics) Student Workshop 2014. Thus, I would like to thank my supervisor Professor Uluğ Bayazıt for giving me valuable advice and support always when needed.

July 2014

Ulukbek Attokurov
(Computer Engineer)

TABLE OF CONTENTS

Page

ABBREVIATIONS

DUC	**:** Data Understanding Conference
HAC	**:** Hierarchical Agglomerative Clustering
IR	**:** Information Retrieval
LCS	**:** Longest Common Subsequence
LSA	**:** Latent Semantic Analysis
LSI	**:** Latent Semantic Indexing
MDS	**:** Multi-document summarization
MMR	**:** Maximal Marginal Relevance
MSE	**:** Mean Squared Error
POS	**:** Parts-of-Speech-Tagging
ROUGE	**:** Recall-Oriented Understudy of Gisting Evaluation
SVD	**:** Singular Value Decomposition

LIST OF TABLES

Page

LIST OF FIGURES

Page

MULTI-DOCUMENT SUMMARIZATION
USING DISTORTION-RATE RATIO

SUMMARY

The present thesis investigates distortion-rate ratio in the context of multi-document summarization. The multi-document summarization is considered as a data compression task. Optimal Tree Pruning algorithm introduced by Breiman et al. and extended by Chou et al. is adapted to multi-document summarization.

The main issue in the multi-document task is redundancy. The input documents discuss the similar topics and thus contain repeated information about them. To avoid the inclusion of the repeated information in the summary, the redundant information have to be detected and eliminated.

Hierarchical Agglomerative Clustering algorithm is used to detect the redundancy in the documents. This algorithm is chosen, since it yields a binary tree that is used in the optimal tree pruning algorithm.

Optimal tree pruning algorithm is employed to reduce the redundancy. An optimal tree that trades off distortion and rate is produced after the pruning. Distortion reflects the semantic loss in the meaning of a sentence if the sentence is represented by another sentence. Thus distortion is adopted as distance between an original sentence and a sentence that represents it. Rate means the amount of information used to present an initial document in a condensed form. Hence, rate is defined to be the number of the sentences included in the document.

λ function, which is the ratio of distortion and rate, is used to determine a sub-tree to be pruned off. A sub-tree yielding the minimal λ value is eliminated, since λ value evaluates increase in distortion for decrease in rate. In each iteration of the pruning, a sub-tree with the minimal λ value is eliminated. The iteration may be stopped when the sufficient number of the sentences are left in the leaf nodes of the tree. In addition, the iteration can be stopped if distortion reaches to a predetermined threshold or the λ value exceeds an optimal value. After pruning step, sentence selection algorithms can be employed to include appropriate sentences in the summary.

Document set to be summarized are represented using different weighting schemes($tf-idf$, tf, 0-1 weighting). Semantic space of the documents is created by using Latent Semantic Analysis that uncovers the semantic relationships between the words.

The proposed system is tested using DUC-2002 data set and evaluated using ROUGE package. The performance of the system is compared with the best systems of DUC-2002 and with the extract based summaries provided by DUC-2002.

BOZULUM-HIZ ORANINA GÖRE ÇOKLU METIN ÖZETININ ÇIKARILMASI

ÖZET

Günümüzde internet ortamında verilerin büyük oranda artması bilgi erişimini zorlaştırmaktadır. Internete erişimi olan herkes metin, görsel ya da işitsel dosyalar yükleyebilir, blog ya da web sitesi oluşturabilir. Dolayısıyla çeşitli türden dokümanlar internet aracılığıyla sanal dünyaya yayınlanmakta ve bilgi kapsitesini arttırmaktadır. Örneğin, Google arama motorunun son iki yılda endekslediği web sitelerinin sayısı 30 milyarı aşmış durumdadır.

Büyük miktarda verilerin arasından gerekli olanlarını bulmak ve en uygununu seçmek zordur. Bazen belli bir konu üzerinde belge araması yaptığımızda arama motorları milyonlarca sonuç üretebilmektedir. İnsanın fizyolojik kapasitesi ve zamanı sınırlı olduğundan milyonlarca dokümanlar üzerinden geçmesi ve uygun olanını seçmesi imkansız ya da zaman alıcıdır.

Yukarıda anlattığımız sorunların üstesinden gelmenin bir yolu doküman özetinin çıkarılmasından geçmektedir. Sanal ortamda bulunan dokümanların büyük bir kısmı metin olduğundan metin özetinin çıkarılması çok sık olarak kullanılan ve araştırılan konulardan bir tanesidir. Metin özeti orijinal dokümanda anlatılan esas konuları kapsar ve onunla ilgili detayları içerir. Metin özeti orijinal dokümanın kullanıcının bilgi ihtiyaçlarını karşılayıp karşılamadığını kısa sürede belirlemesine yardımcı olur.

Tek dokümanın özetini çıkarma yönteminde giriş olarak bir doküman kullanılır. Çoklu doküman özetlemesinde birden fazla dokümanın özeti çıkarılır. Özetleme sistemleri genel ve sorguya dayalı olarak da ikiye ayrılır. Genel özetler orijinal dokümanla ilgili esas konuları ve onlarla ilgili detayları içerir. Sorguya dayalı özetler ise aranan sorguya uygun bilgileri içerir. Kullanıcı sorgusu özetin oluşturulmasında izlenmesi gereken esas kural olarak kullanılır.

Özetleme sistemleri çıkarımsal ya da soyutlayıcı özetleme sistemleri olarak sınıflandırılır. Çıkarımsal özetlemede önemli bilgi kapsayan cümleler seçilerek özet oluşturulur ve cümleler üzerinde hiç bir değişiklik yapılmadan özetleme yapılır. Soyutlayıcı özetleme sistemlerinde ise mevcut sistemler üzerinde değişiklik yapılır ya da yeni cümleler oluşturulur. Bu yüzden soyutlayıcı özetleme sistemleri çıkarımsal özetleme sistemlerine göre karmaşık işlemler gerektirir.

Çoklu metin özetleme sistemlerinde esas amaç bilgi tekrarlanmasının önlenmesidir. Giriş olarak kullanılan dokümanlar aynı konu hakkında yazıldığından benzer metin birimleri(cümleler, paragraflar vb.) doküman kümesi boyunca sık olarak kullanılırlar. Tekrarlanan metin birimleri önemli konuları belirlediği gibi özetlerde eklendikleri zaman bilgi tekrarına yol açarlar. Böyle durumların önlenmesı için benzer metin

birimlerinin belirlenmesi ve onların özette çok sayıda tekrarlanmasının önlenmesi gerekir.

Bilgi tekrarının önlenmesi için bir kaç yöntem geliştirilmesine rağmen bu alanda araştırmalar günümüzde de devam etmektedir. Aynı problem bu bitirme çalışmasında da ele alınmıştır. Bu çalışmada ağaç budama algoritmasının(Optımal Tree Prunıng algorıthm) HAC(Hierarchical Agglomerative Clustering) algoritması ile beraber kullanımı araştırılmıştır. HAC algoritması tekrarlanan metin birimlerinin ayıklanmasında ve ağaç budama algoritması tekrarlanan metin birimlerinin özet metinde azaltılmasında kullanılmıştır.

Orijinal dokümanlar cümlelere ayrıştırıldıktan sonra cümleler HAC algoritması aracılığıyla demetlere atanır. Benzer cümleler aynı demette yer alır. HAC algoritması ağaç yapısında demetler oluşturduğundan cümleler ağacın yapraklarında yer alır. Her bir düğümde temsilci cümleler saklanır. Her bir düğüm için temsilci cümle atanır ve temsilci cümle alt ağacın yapraklarında yer alan cümleleri temsil eder. Ağacın kök düğümünde tüm cümleleri temsil eden temsilci cümle saklanır.

Tekrarlanan metin birimlerinin elimine edilmesi için ağaç budama algoritması kullanılır. Ağaç budama algoritmasının kullanılması için bozulum(distortion) ve hız(rate) parametrelerinin belirlenmesi gerekir. Bir cümle temsilci cümle ile temsil edildiği zaman bilgi kaybına uğradığından bozulum ortaya çıkar. Bozulum bir cümle temsilci cümle ile temsil edildiği zaman ortaya çıkan bilgi kaybı oranını gösterir. Hız ise özeti oluşturmak için kullanılan cümle, kelime ya da harf sayısını gösterir.

İki vektör arasındaki aralık bozulum ölçütü olarak kullanılabilir. İki vektör arasındaki aralığı ölçmek için benzerlik katsayıları kullanılabilir. Kosinüs benzerlik katsayısı en yaygın olarak kullanılan benzerlik katsayılarından bir tanesidir. Kosinüs katsayısı hesaplamaları kolaylaştırdığı gibi bazı problemleri de ortaya çıkarır. Kosinüs katsayısı iki vektörde de yer alan benzer kelimelerin sayısı ve sırasına göre iki vektörün benzerliğini değerlendirir. Dolayısıyla cümlelerin anlamsal benzerliği göz ardı edilir.

Gizli Anlamsal Analiz metodu kelimeler ya da cümleler arasındaki ilişkilerin belirlenmesi için kullanılabilir. Bu yöntem kelimelerin beraber kullanılma istatistiklerine dayanmaktadır. Benzer konuların anlatılmasında benzer kelimeler ve belli kalıplar kullanılır. Benzer kelimelerin ve kalıpların belirlenmesi metin parçası içindeki anlamsal ilişkilerin belirlenmesine yardımcı olur. Kelimelerin başka kelimeler ile olan ilişkilerine ya da kalıplar içerisinde kullanımına bakılarak ağırlıklandırılması metin birimlerinin benzerliklerinin belirlenmesinde önemli rol oynar.

Ağaç budama algoritması bozulumu minimize eden ve hızı azaltan alt ağaçları elimine eder. Ağacın yaprak düğümlerinden kök düğümüne doğru ilerledikçe bozulum da artış izlenir ama veri sayısı azalır. Yaprak düğümlerinde bozulum sıfıra eşitken ağacın kök düğümünde en büyük değerine ulaşır. Bu yüzden mevcut alt ağaçların içinden bozulum ve veri sayısı oranını minimize eden alt ağaç budanır.

Özetleme 4 aşamadan oluşmaktadır. Birinci aşamada metin ön işlemesi yapılır, metin cümlelere ayrıştırılır ve cümleler vektör olarak gösterilir. Kelimenin kökünün bulunmasından kelime - cümle matrisinin oluşturulmasına kadar olan işlemler bu aşamada yapılır. Ikinci aşamada ise tekrarlanan metin birimlerini belirlemek için cümlelerin demetlenmesi yapılır ve benzer cümleler aynı demetlere atanır. HAC

algoritması ağaç yapısında veri yapısı ürettiğinden cümleler HAC ağacında saklanır. Bir birine benzeyen cümleler aynı alt ağacın yaprak düğümlerinde yer alır. Benzerlik ölçütü olarak kosinüs benzerlik katsayısı kullanılır. Aynı cümlelerün benzerlik katsayısı bire eşittir, ama birbirine tamamen benzemeyen cümlelerin benzerlik katsayı derecesi sıfıra eşittir.

Üçüncü aşamada ise tekrarlanan metin birimleri elimine edilir. Bir önceki aşamada elde edilen ağaç üzerinde ağaç budama işlemi gerçekleştirilir. Her bir budama iterasyonunda bozulum ve veri sayısı parametrelerine göre alt ağaçlar budanır. Alt ağaçlar benzer cümleleri kapsadığından budama sonucu benzer cümleler temsilci cümleyle değiştirilmiş olur. Dolayısıyla bilgi tekrarlanması problemi giderilmiş olur. Budama işlemi cümle sayısına ya da bozulum değerinme göre durdurulabilir. Iterasyonun durdurulması için hangi parametrenin kullanılacağı sistemin özelliklerine göre ayarlanır.

Dördüncü aşamada ise özet oluşturulur. Özet oluşturmak için bir önceki aşama sonrası elde edilen alt ağaç kullanılır. Alt ağacın yaprak düğümlerinde yer alan cümleler kullanılarak özet oluşturulur. Özet ise yaprak düğümlerde yer alan cümlelerin hepsinden oluşturulabilir ya da cümle seçme algoritmalarından yararlanılarak cümlelerden bazıları seçilebilir.

Sistem performansı ROUGE paketi kullanılarak değerlendirilmiştir. ROUGE paketi model ve sistem özetlerini birbirine karşılaştırır. Sistemin performansı DUC-2002 veri seti kullanılarak test edilmiştir. DUC-2002 veri seti Data Understanding Conferance isimli konferans için hazırlanmıştır. Veri seti konferansa gönderilen sistemlerin test edilmesi için kullanılmıştır. Veri seti 59 dokümandan oluşmuş ve dokümanlar 4 sınıfa atanmıştır. Her bir doküman seti 200 ve 400 kelimeden oluşan soyut ve çıkarımsal örnek özetleri de içermektedir. Örnek özetler konferans tarafından hazırlanmıştır. Çıkarımsal özetler 400 kelimelik uzunluktaki özetle, soyut özet ise 200 kelimelik özetle karşılaştırılmıştır. Örnek özette bulunan cümlelerden çok sayıda kapsayan ve sistem tarafından üretilen özet başarılı özet olarak kabul edilmiştir.

Önerilen sistemin değerlendirilmesinde iki test senaryosu izlenmiştir. Birinci senaryoda 200 kelimelik soyut özet kullanılmıştır. Sistem tarafından üretilen özet(aday özet) 200 kelimeden oluşan örnek özetle karşılaştırılmıştır. Sistemin performansını değerlendirmek için Rouge-1 Precision, Rouge-1 Recall, Rouge-1 F1 ölçütleri kullanılmıştır. Rouge-1 kullanıldığında iki özet(aday ve örnek özetler) kelime bazında değerlendirilmiş olur. Rouge-1 Recall değeri aday özetin örnek özette bulunan kelimeleri ne kadar içerdiğini gösterir. Bu yüzden Rouge -1 Recall değeri aday özet ile örnek özet arasındaki benzerlik oranını gösterir.

İkinci senaryoda ise 400 kelimedenşan örnek özet kullanılmıştır. Sistem performansı aday ve örnek özette bulunan cümleler bazında değerlendirilmiştir. Sentence Recall ve Sentence Precision ölçütleri aracılığı ile sistem performansı ya da özet içeriği değerlendirilmiştir. Sentence Recall aday özette bulunan örnek özet cümleleri sayısının toplam örnek özet cümle sayısına göre oranını gösterir. Sentence Presicion ise aday özette bulunan örnek özet cümlelerinin toplam özet cümle sayısına göre oranını belirler.

DUC-2002 konferansında en iyi sonuç gösteren sistemlere göre önerilen sistem performansının daha başarılı olduğu tespit edilmiştir. Bir sonraki araştırmada ise önerilen sistem soyut özetlerin oluşturulması için kullanılacaktır.

1. INTRODUCTION

Nowadays, the massive amount of information available in the form of digital media over the internet makes us seek effective ways of accessing this information. Textual documents, audio and video materials are uploaded every second. For instance, the number of Google's indexed web pages has exceeded 30 billion web pages in the last two years. Extraction of the needed information from a massive information pool is a challenging task. The task of skimming all the documents in their entirety before deciding which information is relevant is very time consuming.

One of the well known and extensively studied methods for solving this problem is summarization. Text summarization produces a short version of a document that covers the main topics in it [19]. It enables the reader to determine in a timely manner whether a given document satisfies his/her needs or not.

A single document summarization system produces a summary of only one document whereas a multi-document summarization system produces a summary based on multiple documents on the same topic. Summarization systems can also be categorized as generic or query-based. A generic summary contains general information about particular documents. It includes any information supposed to be important and somehow linked to the topics of the document set. In contrast, a query based summary is comprised of information relevant to the given query. In this case, query is a rule according to which a summary is to be generated.

Summarization systems can be also classified as extractive or abstractive. In extractive systems, a summary is created by selecting important sentences from a document. Here, only sentences containing information related to the main topics of the document are considered to be important. These sentences are added to the summary without any modification. On the other hand, abstractive systems can modify the existing sentences or even generate new sentences to be included in the summary. Therefore, abstractive summarization is typically more complex than extractive summarization.

The main goal in multi-document summarization is redundancy elimination. Since the documents are related to the same topics, similar text units(passages, sentences etc.) are encountered frequently in different documents. Such text units that indicate the importance of the topics discussed within them should be detected in order to reduce the redundancy. Some of the well-known approaches that address this problem are briefly explained in the following section.

Although much work has been done to eliminate the redundancy in multi-document summarization, the problem is still actual and addressed in the current work as well. The current work proposes to integrate the generalized BFOS algorithm [8] adopted by Chou et.al [10] for pruned tree structured quantizer design with the HAC (Hierarchical Agglomerative Clustering) algorithm. The two main parameters (distortion and rate) in the latter work are adopted to the multi-document summarization task. Distortion can be succinctly defined as the information loss in the meaning of the sentences due to their representation with other sentences. More specifically, in the current context, distortion contribution of a cluster is taken to be the sum of the distances between the vector representations of the sentences in the cluster and the vector representation of the cluster. Rate of a summary is defined to be the number of sentences in the summary, but more precise definitions involving word or character counts are also possible. BFOS based tree pruning algorithm is applied to the tree built with the HAC algorithm. HAC algorithm is used for clustering purposes since BFOS algorithm gets tree structured data as an input. It is found that the suggested approach yields better results in terms of the ROUGE-1 Recall measure [31] when compared to 400 word extractive summaries(400E) included in the DUC-2002 data set. Also, the results with the proposed method are higher than the ones obtained with the best systems of DUC-2002 in terms of sentence recall and precision [15]- [16].

1.1 Literature Review

Goldstein et al. [14] proposed a measure named Maximal Marginal Relevance(MMR) which is used to detect redundant sentences. The system produces an extract based summary relevant to the query of a user. MMR minimizes the redundancy while maximizes the relevancy of the summaries. The system is designed following the

general scheme described below. In the first stage, the text is parsed into sentences. The sentences are interpreted with the bag of words model and are represented in the Vector Space. In the next stage, the similarity between a passage and a query is calculated. The passages with similarity below the predefined threshold are eliminated. Since cosine similarity is used, similarity calculation is based on the word overlap. In the last stage, MMR measure is applied to determine the passages salient for the summary. The passages relevant to the query, but dissimilar to other passages already contained in the summary are selected and ordered. The ordering is done following some criteria like the order in the text or time of creation.

Lin et al. [33] used different approaches to single document summarization and developed a system named NeATS. In this system, important topics are determined first and the sentences are weighted according to the correlation with the main topics. Summary worthy sentences are defined using the following parameters: position of the sentence, stigma words and MMR.

Radev et al. [42] developed a system called MEAD based on statistical methods. The centroid vector for the given document set is determined. It contains the words related to the main topics of the source and is used to determine the sentences somehow linked to the main content of the documents. The similarity to the centroid, the position in the document, the word overlap with the first sentence and the word overlap with other sentences are calculated for each sentence.

Barzilay and Elhadad [3] built a system based on the relations of the words. They used WordNet thesaurus to determine the relationships(synonymy, holonymy etc.) between the words. Lexical chains are built by using nouns and noun compounds. The words are included into the chains by WordNet relations of their meanings. In the following stages, the chains are weighted and the sentences containing the strong chains are selected to be included in the summary. Word count and word overlap are used to rank the lexical chains and to find the appropriate sentences for the summary.

Barzilay and McKeown [38] approached to the summarization task in different manner. Their proposed system produces the summary by generating new sentences instead of

using existing ones. The cluster of sentences are created in order to infer the common clauses from the sentences in each cluster.

In the following works different approach to the text summarization is implemented. Clustering of the text passages(sentences, words etc.) is used for the summarization purposes. The system developed by Seno and Nunes [48] clusters the sentences incrementally; a certain number of sentences are assigned to an appropriate cluster in each iteration of clustering. Initially, the first sentence of the first document forms the first cluster. In the next steps, the following sentences are included to the existing clusters if they meet certain requirements. Two methods of similarity measure is tested in the system. In the first case, word overlap between a candidate sentence and the cluster is used as a similarity measure. The ratio of the common terms to the number of the total terms in the candidate sentence and the current cluster is used as a similarity measure. A candidate sentence is assigned to a current cluster if the value of the word overlap is greater than the predefined threshold. The most optimal threshold found is 0.2; if the word overlap value is less than 0.2 for each existing cluster then a new cluster that contains the candidate sentence is created. In the second case, the cosine similarity is used as a similarity measure. Cosine similarity is calculated between a candidate sentence and the centroid of the current cluster. The centroid is made up of words which conveys most of the meaning about the topics of the documents. Which term to include in the centroid is decided using by means of statistical weighting schemes. TF-IDF and TF-ISF weighting schemes are used to determine the topic related terms. The best clustering results are achieved when TF-IDF is used for the term weighting purposes.

Hatzivassiloglou et al. [17], [18] created a system called SimFinder. It is incorporated to the multi-document summarizer system proposed by McKeown et al. [38] that uses text reformulation for abstract generation as described in [4]. SimFinder is based on the clustering of the sentences. Clustering is not hierarchical and is used to group similar sentences that share common information about the topics in the text. The similarity between the sentences or paragraphs is calculated using primitive and complex features. Primitive features are made up of single words, word co-occurrence, noun phrases, WordNet synonyms etc. and complex features are made up of pairs of

primitive features. Each cluster is represented with a representative sentence included in the summary. MultiGen system is used to generate a representative sentence from the common information contained in the sentences assigned to the same cluster.

In recent years, algebraic methods are used widely for text summarization purposes. One of the most important algebraic tool is LSA(Latent Semantic Analysis) [29]. LSA based algorithms are used to decrease the dimension of the data set, to unearth the semantic relations between the concepts or the documents or to represent data samples in Semantic Space. In addition, text summarization tasks can be carried out using LSA based algorithms.

In the context of text summarization, SVD serves as a tool that captures the relationships between the terms. SVD determines the relationships between the words using co-occurrence statistics of the terms and the word usage patterns. Moreover, terms and sentences are projected into the same semantic space and are represented in that space. That is why, terms and sentences can be clustered and they are can be compared to each other.

It is supposed that each row of D^T corresponds to the topic or to the word usage pattern in the text and their corresponding singular values indicate the importance degree of the topics. Hence, the summary worthy sentences may be determined by calculating the length of appropriate vectors in S^2D^T.

Bing et al. [6] developed a system based on clustering and LSA. Term-to-sentence matrix is decomposed using SVD. The sentences with the highest similarity is determined. The most similar sentences are combined to create a new sentence which is called a fake sentence. This sentence is longer than the other sentences in the set. Term-to-sentence matrix is updated taking into account a newly created fake sentence. The sentences used in creating a fake sentence are excluded from the set of sentences. Again the most similar pair of sentences is calculated and it is merged by yielding a fake sentence. These procedures are repeated until predefined number of sentence clusters are obtained. Each sentence cluster is represented by the centroid sentence. Finally, a summary is generated using the centroid sentences.

Steinberger and Krist [50] dealt with the multi-document summarization task in the context of LSA. They applied LSA to single document summarization and adapted the developed system for multi-document summarization. The summarization system starts by creating term-to-sentence matrix A where rows represent the terms and columns represent the sentences. The cells are filled with the TF-IDF weighting scheme. In the next steps, term-to-sentence matrix A is decomposed into three matrices T, S and D^T by applying SVD(Singular Value Decomposition). The matrices are made up of r linearly independent base vectors. The sub matrices S and D^T are used to create the ranking matrix SD^T which is used to determine the salient sentences for the summary. Each column vector of SD^T is ranked according to its length and the resultant ranking is used to determine which sentences are included in the summary. Each top ranked sentence is included to the summary if the corresponding vector has the highest score and the candidate sentence is not similar to sentences already contained in the summary. Score equals to the length of the vector divided by the number of the terms contained in that vector. The similarity between the candidate sentence and a sentence in the extract is calculated with the cosine similarity measure in initial term space.

2. MULTI-DOCUMENT SUMMARIZATION

Most summarization systems include three modules: analysis, processing and generation. In the first stage linguistic and lexical analysis are performed. This may include parsing the paragraphs, sentences or words as well as stemming and stop word elimination. Also term-sentence matrix is created here. Further processing operations try to determine the redundant information in the documents. The repeated information is considered as redundant in MDS(Multi-document summarization). Redundancy is the main property of the input, since the documents in the set are written about the same topic. To determine redundancy, various statistical and linguistic methods can be used.

In the next stage, redundancy elimination is performed. The redundant information should be eliminated since the main purpose of the MDS is to present the summary in a condensed manner without repeating the same content. Clustering is a simple and widely used approach which can be applied for redundancy elimination. It determines similar lexical units of the text(redundancy detection) and groups them into the same cluster. A representative sentence might be selected from each cluster in order to reduce the redundancy.

The last step is summary generation. The sentences for the summary is selected from the remaining sentences after redundancy elimination has been performed. Different approaches may be followed for the summary generation. For example, one sentence may be included into the summary from each cluster.

2.1 Stages of Summarization

Summarization procedure is decomposed into three stages:
1.input text processing to obtain a text representation(interpretation stage)
2.transforming the source representation into the summary representation(transformation stage)

3.the summary generation(generation stage)

These steps have to be followed carefully to produce the summary efficiently. In addition, context factors should be analysed. In accordance with [22] these factors are classified as input, purpose and output.

Input factors.

The features of the input text play a crucial role in the summarization procedure. In most cases, they determine the path to be followed and the output of the summarization system. The most important aspects related to the summarization task are listed below:

Source text structure: Labels stating the structure of the document like paragraph, sentence, section, chapter can be used in summarization. They may mark the places where the lexical units important for the summary are contained. For instance, in positional based techniques sentences located at the beginning of the paragraph are supposed to be more suitable for the summary. In addition, words included in headings may be assigned more weight in comparison with other words contained in another parts of the text.

Subject: Domain-sensitive systems produce summaries related to the specified domain. This is beneficial if a feature set includes the lexical units that describes the related domain. Moreover, systems destined for the specific domain allow to adopt different Natural Language Processing techniques suitable for the domain under consideration.

Scale: Scale determines the minimum lexical unit. Lexical unit is used in interpretation and in transformation stages as a main building block of summarization. Sentences and even clauses may be used as a minimal textual unit when news articles are processed. However, paragraphs are the correct choice for textual units when long texts are considered in the summarization task.

Unit: A summarization task is categorized as single and multi document summarization. If several documents written about the same topics are summarized then it is called multi-document summarization; otherwise it is named single document summarization.

Purpose factors: In some systems the purpose factors of the system are not stated exactly. Because summarization is considered as condensation of the text. But

task-driven summarization is beneficial since it can be set efficiently to meet the specific requirements of users. For instance, in IR(Informatio Retrieval) systems summarization may be used to create snippets.

Purpose factors can be decomposed into three classes: situation, audience and use.
Situation. This label states the context. Situation may be distinguished as tied and floating. In the first case, the summary is formed under strict requirements. The requirements may define goal of the summary, purpose of the usage, the length of the summary etc.. In floating type, the summary is created without any requirements and specifications. The summary contains the information related to the main topics of the text.

Audience. This factor states the readers for whom the summary is produced. If a reader is a scientist who is interested in special topics in computer science probably he or she needs a summary intended for the computer scientists. Probably, background information is needed to use a summary intended for the special audience. On the other hand, a summary created from the news articles is aimed for general audience. It is created without considering special information needs of a user.

Use. The third purpose factor determines the usage of a summary. Specifically, it refers to the aim of usage. A summary can be used for different goals. It may assist to readers to outline the huge amount of information, to get preview of the materials under consideration or to refresh memory of a user if a user has background information about the summarized texts. Google's snippets are one of the possible examples of the practical uses of a summary.

Output factors
Output factors deal with the output of the summarization system. These factors determine the structure of the summary text, presentation of the content and the style of the text. Thus the main output factors are distinguished as material, format and style.

content: A summary may cover the main features of source text. The relevancy of the lexical units are determined by criteria like statistical, linguistic, positional features. A summary tries to cover the essential topics discussed in the text and gives detailed information about them. A user may have a general overview about the material

under consideration. This kind of summaries are called generic summaries. On the other hand, query-driven summaries contain specific information shaped by the specifications and requirements about the content. The specifications or requirements are related to the information needs of a user. The information need of a user may be presented as a query, keywords or questions. All in all, generic summaries cover all topics considered to be important and related to the main topics of a source text; whereas query-driven summaries contain information relevant to the specified user query.

style: An informative summary outlines a source text. It gives a general overview about the source of the summary. It describes the topics discussed in the text. An indicative summary contains information about the topics in the source documents. It briefly explains the main points of the summarized texts. An aggregative summary provides an additional information non-existing in the original text or may include texts from other sources.

Production process: An extractive summary is produced by selecting the important lexical units. The extracted units constitute the summary without any lexical modifications. Thus the summary is some portion of the source. With an abstractive summary the this is not case: the existing sentences are modified or new sentences are generated to create a summary. The summary is an interpretation of the original text.

length: It is a main property of a summary. If the summary is short and contains the most important parts of the source then it is preferable in many cases. Specially, if the summarization is used for data compression purposes then short summaries are selected to present the original text. In most applications an upper and a lower boundary for the length should be determined. If the summary length is too short then the important information may be lost. On the other hand, if the summary is too long, a summary may contain noisy sentences. Noisy sentences do not carry essential information about the main topics of the topic so they have to be eliminated from the summary.

2.2 The Main Classification of the Text Summarization Approaches

There are many methods used for text summarization. Each method is based on the different characteristics of the text under consideration. Some of them use statistical information of the lexical units while other methods take advantage of the linguistic features of the text. Depending on the features used in determining the lexical units to be transferred into the summary and relationships between the text segments, the main approaches can be classified as surface level, entity level and discourse level. Recently, there appeared new methods involving the corpus based statistics, algebraic and graph based approaches. The main classifications are described and discussed in the following sections.

2.2.1 Surface level approaches

This approach uses the surface features to determine the salience function. Features are examined to decide which lexical units to include in the resulting summary. Luhn [34] used term frequency to extract relevant text portions. The idea behind the method is based on the assumption that the most salient sentences uses the most frequent words in the text. It is supposed that authors tend to use the words related to the main topics frequently. The score is calculated using appropriate saliency function which correctly reflects the significance of the sentence properly. The sum of the term frequencies of the terms contained in the sentence can be used as a salience function.

Another text characteristic which may be used as an indicator of the important lexical units is their location in the source [5], [7]. Words contained in the heading or at the beginning of the paragraphs may be assigned greater weight than the rest. Additionally, the first sentences in the paragraphs can be included in the summary since they inform a reader about the main topic of the paragraph. Furthermore, sentences from each paragraph may be transferred to the summary depending on the location of the paragraph in the document. The number of the sentences to be included from the first and the last paragraphs may be greater. Edmundson [12] combined cue words, title words and positional information to extract the most relevant lexical units. He showed that the combination of these characteristics produces a summary close to abstract summary created by a human.

Cue words or phrases are another type of the indicators which determine the most relevant lexical units. Phrases like "All in all", "in conclusion" etc. signal the end of the opinion discussed in a text or in a paragraph; therefore text portions containing these phrases may be included in the summary, since the whole topic is summarized in the conclusion of the text or the paragraph. Due to the explicit meaning and the role of the cue words in the text, the task of determining the text units to be included in the summary becomes easier compared with other methods where statistical or algebraic calculations have to be done to assign a priority to the lexical units.

2.2.2 Text connectivity or cohesion based approaches

Another approach for text summarization is the text connectivity based approach. Linking to the precedent parts of the text is one of the main ideas of the method. It uses relations between expressions and concepts in the text. Methods dealing with lexical chains and Rhetorical Structure Theory are known representatives of the text connectivity approach.

Lexical chains uses cohesive relations(synonymy, holonymy etc.) between terms. The semantic relations are determined by means of WordNet and dictionaries. Lexical chains are constructed using semantic relations between the terms. The number of an element in a chain and their type determine the score of the sentences. Sentences with concentrated strongest chains are selected to be included in the summary.

Rhetorical Structure Theory(RST) is another type of the text connectivity based method. It builds a tree representing the structure of the text. The relations link nucleus(the central part of the text) with a satellite(less central part of the text). Nucleus text units are weighted with 1; whereas satellite units are assigned 0 value. A score of a sentence is evaluated by the sum of the scores found on the nodes from the root node to the sentence node. [39], [35] are examples of the text summarization approaches using RST.

2.2.3 Corpus-based statistical approaches to summarization

This type of methods use statistical properties of the text units. It is assumed that the main topics of the documents are related to lexical units with certain statistical properties. For example, authors generally tend to use topic related words or lexical constructions frequently. These words have to be discriminated from the other words in order to obtain more accurate weighting scheme. To this end, term frequency can be used which is the frequency of the occurrence of the term in the current document. Despite the fact that term frequency helps to increase the recall of the retrieval, but it may lead to the retrieval of the non relevant items which causes the decrease of precision. Specifically, if high frequency terms are not concentrated in a few documents but instead they are prevalent in the whole collection, it is possible that the non relevant documents containing frequent terms may be retrieved. For instance, the word "Economic" may be prevalent in the collection of documents written about economy, finance, management etc. and it has not descriptive property in order to group the documents into clusters as finance, management etc. To prevent such kind of problems, it is needed to obtain a weighting scheme that takes into account the statistical properties of the whole collection of the documents. Inverse document frequency is used to perform this task.

It is shown that the importance of a term is inversely proportional to the number of documents in which the term is contained(document frequency) [49]. Inverse document frequency is used to reflect the dependency on the document frequency. idf assigns greater weights to the terms with the less document frequency value; the terms included in a few documents are important terms compared to the terms contained in all documents.

Inverse Document Frequency combines term frequency and inverse document frequency. The term with the highest term frequency and the lowest document frequency is considered to be the most important term which distinguishes the relevant documents from the other documents in the collection. Hence, this suggests that a reasonable weighting scheme has to consist of two main components: term frequency and inverse document frequency. The final formula is given as $tf * idf$ where tf is the term frequency and idf is inverse document frequency.

Finally, the sentences may be scored by applying different strategies depending on the specifications of the problems under consideration. The most simple one is the sum of the weights corresponding to the terms included in the sentence. The top ranked sentences are selected to create a summary.

Summarist system [20] is based on the statistics of the concepts. It counts concepts instead of the words. Concept generalization is employed to identify a general concept summarizing other linked concepts. The relation between the concepts is determined by using WordNet. Occurrence of a word linked to the general concept increases it's frequency. For instance, the counter for the concept "computer" is incremented when notebook or desktop computer is found in the text.

2.2.4 Graph based approach

Graph based algorithms like Google's page rank, HITS have been applied in many areas. They have been used in social networks, in citation analysis and in analysis of the link-structure of WEB [51]. Also this type of algorithms are used in text summarization.

The vertices of the graph represent the sentences and the edges show the similarity between the sentences. The content overlap(the number of common words or tokens, or overlapped phrases) can be used as a similarity measure between sentences. The similarity defines the degree of connection between nodes. The higher the similarity the stronger the linkage among the connected nodes. After the graph is built, a ranking algorithm is employed. Finally, the top ranked sentences are included in the summary.

2.2.5 Algebraic approaches

LSA based text summarization is widespread in recent years. LSA helps to infer the main topics in the documents to be summarized. It decomposes the term-document matrix into U, S, V matrices where U represents the terms in the semantic space, V corresponds to the documents and S shows the importance degree of the topics in the text. Since the terms and documents are presented in the same space with equal number of dimensions, term to term, term to document, document to document similarities can

be calculated. The examples for the text summarization using LSA are the works of [51], [30], [27].

2.3 Evaluation Measures

The evaluation of a summary is an important part of the text summarization task. The main goal of the evaluation measure is to determine whether the summary captures the main content of the original document. It shows the quality of the produced summary.

The quality of the summary may be assessed using different approaches. The most reliable method for evaluating the summary is human judgement. Since annotator can determine the difference between the original and the summary according to the topics discussed in both texts. He/she can also decide whether the summary may be used instead of the summarized documents or not. Text summary may be assigned a score from the predefined scale depending on the judgement of the annotators.

One of the intrinsic evaluation methods is content based evaluation. It is performed by comparing the candidate summary produced automatically and the ideal summary written by a human. If a summary is similar to the ideal summary then it is considered to be of high quality. Co-selection evaluation is used in extract based summarization. The number of the ideal sentences found in the produced summary determines the value of the co-selection measure. Another evaluation measure is task-based method. It evaluates how much the summary conforms to a certain task.

2.3.1 Text quality measures

Text quality is determined using several criterions:
-grammaticality- the text should not contain any grammatically incorrect items(sentences, words, punctuation errors)
non-redundancy- the text should not include repeated information as well as similar lexical units(sentences, passages)
coherence and structure- the sentences, passages should be organized accurately. Sentences should be connected and in correct time or logical sequence, the sentences

in their entirety should discuss the main topics or ideas of the summarized document without destroying the structure of the text.

2.3.2 Co-selection measures

F-score, precision, recall are used in co-selection evaluation. Precision shows the proportion of the relevant sentences retrieved. Here relevancy means the occurrence of a sentence in the ideal summary. Hence, precision is the number of the sentences that occur in both ideal and automatic summaries divided by the number of the sentences found in the automatic summary. Recall shows the proportion of the relevant but not retrieved sentences. It may be calculated as the ratio of the number of the sentences included in both the ideal and system summaries to the number of the sentences in the ideal summary. F-score is the harmonic measure which takes into account the precision and recall in evaluation of the summary. The most basic form of the F-score is the $F1$ measure which is the harmonic average of precision and recall:

$$F1 = 2 * \frac{P * R}{P + R} \tag{2.1}$$

In some cases more complex forms of F score that use a function of β parameter may be used and this is defined as below:

$$F = (\beta^2 + 1) * \frac{P * R}{\beta^2 * P + R} \tag{2.2}$$

,where β is a constant factor which increases the value of precision when $\beta > 1$ and favours recall when $\beta < 1$.

2.3.3 Content-based measures

As shown in preceding section, co-selection measures evaluates the proprotion of the exactly matching sentences. However, in reality different authors tend to use different lexical constructions and words to express the same concept or event. The order or the words used in the sentences may differ but the meaning or the main idea of the sentence stays the same. Thus the sentences differing from each other by their lexical content or grammatical structure should be evaluated accurately by taking into account the similarity of their meanings. To overcome the discussed problem, several measures operating on the word level are proposed.

2.3.4 Cosine similarity

The most well-known and widely used in IR content-based similarity measure is Cosine Similarity [13]. It is built on the word or token overlap between the given sentences X and Y. The order of the words affects the resulting value of the similarity. The formula for calculating cosine similarity is given below:

$$cos(X,Y) = \frac{X * Y}{\|X\| * \|Y\|} \tag{2.3}$$

where, X and Y are the two vectors whose similarity is evaluated.

2.3.5 Unit overlap

Another measure which is based on the overlap of the lexical units is unit overlap. It evaluates the proportion of the same tokens that are the same without considering their order in the sentence. The formula for calculating unit overlap is given below:

$$unitoverlap(X,Y) = \frac{\|X\| \bigcap \|Y\|}{\|X\| + \|Y\| - \|X \bigcap Y\|} \tag{2.4}$$

,where X and Y are the sets of the tokens or lexical units. $\|X\|$ is the number of the tokens or the lexical units contained in the set X.

2.3.6 Longest common subsequence

Longest Common Subsequence(LCS) [44] is another type of the content-based evaluation measure.

$$lcs(X,Y) = \frac{length(X) + length(Y) - edit_{di}(X,Y)}{2} \tag{2.5}$$

where X and Y are represented as a sequence of tokens or lexical units. $length(X)$ is the length of the string X and $edit(X,Y)$ is the edit distance between two strings X and Y.

2.3.7 ROUGE co-occurrence statistics

ROUGE statistics is built on the matching and co-occurring n-grams. It calculates recall using co-occurred n-grams. Thus it may be named recall based co-occurrence measure. Rouge-n statistics are evaluated using one or more reference summaries which were mentioned as ideal summaries in preceding sections. It counts the number of matching n-grams occurring in the candidate summary produced by the summarization system and the reference summaries. It should be noticed that all reference summaries are taken into account in counting co-occurring n-grams.

$$ROUGE-n(Recall) = \frac{\sum_{C \in RSS} \sum_{gram_n \in C} Count_{match}(gram_n)}{\sum_{C \in RSS} \sum_{gram_n \in C} Count(gram_n)} \quad \textbf{(2.6)}$$

where $count_{match}(gram_n)$ is the maximum number of matching occurrences of n-gram $gram_n$ and $count(gram_n)$ is the number of the n-gram $gram_n$ in the reference summaries. Also in a similar manner, n-gram based precision and F-score may be calculated easily.

2.3.8 Task based measures

This type of measures evaluate to what extent the summary accomplishes the predefined task. In this approach, created summaries are evaluated according to their fulfilment of the given task. The summary is considered to be useful for a system when it suits the purpose of the system. Task based evaluation may be considered under different tasks in various areas.

3. METHODS

3.1 Term Weighting

Term weighting determines the importance of the terms. It can be used to rank terms, sentences, phrases, passages or any text unit in the text. In addition, it is a way to distinguish the terms related to the query or to the topic of the documents.

The simplest approach in term weighting is to assign 1 or 0 depending on the occurrence of that term in the text unit. However, it is difficult to order the terms according to their importance in such type of weighting. Terms can be classified into two classes: term occurs or not occurs. Instead, the number of occurrences of the term in the text unit can be used as a weighting scheme. This type of assigning weight to the term is called term frequency and it is denoted as tf. It shows the importance of terms more accurately than the previous approach. If one term is repeated many times it may be the key word that relates to the main topic or idea of a text unit. But tf does not show the real weight of the term if it is evaluated in the context of the entire collection written about the same topic. tf describes a term within boundary of a single document. Hence, it is a local weighting scheme.

As stated in previous paragraph, term frequency(tf) does not discriminate the relevant words. Because all words are considered to be equally important. In relevancy determination inverse term frequency is introduced in order to depress the effect of words which occur too often in the collection of documents .

Inverse document frequency is the fraction of the total number of the documents, N, in a collection to the document frequency(df) of a term t, where df is the number of the documents containing the term t. idf of a term t is formulated as follows:

$$idf = N/df \tag{3.1}$$

Thus the weight of a rare term is higher, whereas the weight of a term occurred in many documents is lower. It is called global weighting scheme, since idf is calculated by taking into account all documents in a collection,.

Alternatively, term frequency(tf) and Inverse Document Frequency(idf) may be combined to assign a weight to a term t. It evaluates a term according to its frequency in the current document d and scales the weight depending on the occurrence of the term t in other documents in a collection. The $tf-idf$ weight of a term t in document d is defined as

$$tf-idf_{t,d} = tf_{t,d} idf_t \tag{3.2}$$

,where $tf_{t,d}$ is term frequency of the term t in document d and idf_t is inverse document frequency of the term t. In other words, the weight given to a term t varies as described below:

1. the weight is the highest if the term occurs in the small number of documents and it is repeated frequently in the current document d;

2. the weight is assigned a lower value when the term is used a few times in the current document d, but it occurs in many documents;

3. the weight is assigned the lowest value if the term frequency gets the smallest value in the current document d and it is used in all documents in a collection.

In the current study, the modified version named $ntf-idf$ is used where ntf is the normalized term frequency. It is shown that the using $ntf-idf$ improves the result of retrieval [21]. ntf is a fraction of term frequency tf of a term t in a document d and the maximum term frequency max $tf_{j,d}$ in the document d.

$$ntf_{t,d} = tf_{t,d}/max\, tf_{j,d} \tag{3.3}$$

In sentence clustering, idf is renamed to isf(inverse sentence frequency), since the sentences are involved in clustering. Thus the weighting scheme is modified as shown below:

$$isf-ntf_{t,d} = ntf_{t,d} * isf_t \tag{3.4}$$

,where $ntf_{t,d}$ is the normalized term frequency and isf_t is the inverse sentence frequency.

3.2 Similarity Measure

3.2.1 Dot product

As stated in previous sections, a sentence is represented by its vector. The vector of a sentence S is denoted as $V(S)$ and for each component of $V(S)$ there corresponds a term from the feature set. Each cell of the vector is filled with a weight, for instance $NTF - ISF$, of an appropriate term. The set of sentences can than be transferred to the vector space where each dimension represents term. Figure 3.1 shows the sentence

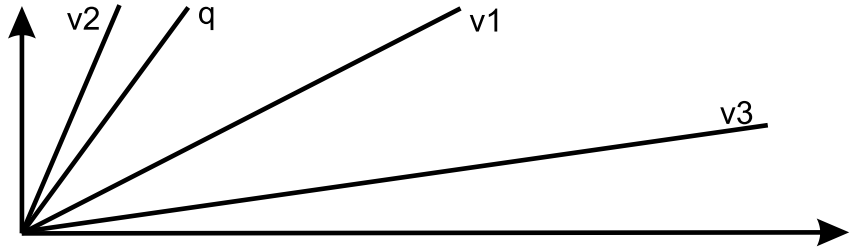

Figure 3.1: A sentence and a query representation in Vector Space.

vectors($v1, v2, v3$) and a query vector(q). As shown, $v2$ is close to the query vector. Moreover, the angle between q and $v2$ is small. This observation can be used in the calculation of the similarity between the sentences and the query. The larger an angle is, the lower the similarity between a document and a query. If two documents are the same, their corresponding vectors match and points to the same point in the vector space. Furthermore, the angle between them is equal to zero which indicates the maximum similarity. Thus, a way is needed to represent the similarity by using the an angle between the vectors. To this end, cosine similarity is used in many IR applications.

The cosine similarity evaluates the similarity between the given sentences $v1$ and $v2$.

$$cos(v1, v2) = \frac{v1 * v2}{\|v1\| * \|v2\|} \tag{3.5}$$

,where the numerator is the dot product of the given vectors and the denominator is the product of their Euclidean lengths. The dot product of $v1$ and $v2$ is the sum over products of the corresponding pair-wise components of $v1$ and $v2$.

3.2.2 Distance metrics

Distance captures the difference between two given objects. It is inversely related to similarity. While similarity determines the similar behaviour of the observations, the distance shows the unlikeness of the observations. If similarity measure is denoted as *sim* and it ranges between 0 and 1, then the distance may be defined as $1 - sim$. Consequently, the more the similarity between the observations, the lower the distance between them.

Distance metrics satisfy the following properties:
1. Non-negativity: $d(i, j) >= 0$ (Distance is not a negative).
2. Identity of indiscernibles: $d(i,i) = 0$ (The distance between same observations is 0).
3. Symmetry: $d(i, j) = d(j,i)$ (The distance does not change if the arguments are reordered).
4. Triangle inequality: $d(i, j) <= d(i,k) + d(k, j)$ (The distance between observations i and j is no more than the one calculated over the observation k).

There are many distance measures used in practice. The most simple and widely used one is based on the similarity measures described above. Specifically, cosine, Pearson etc. similarity measures can be used to determine the distance. if $cos(v1, v2)$ is the similarity of $v1$ and $v2$ then $1 - cos(v1, v2)$ is the distance between them. Pearson correlation coefficient can be also used instead of the cosine similarity. Pearson Correlation Coefficient determines the dependency or relatedness of the random variables X and Y and determined as

$$r = \frac{n(\sum xy) - \sum x \sum y}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}} \tag{3.6}$$

,where r is Pearson correlation coefficient, x is values in the first set of data, y is values in the second set of data, n is the total number of values.

Another popular distance measure is Euclidean distance. Let $i = (x1, x2, ...xn)$ and $j = (y1, y2, ..., yn)$ be two observation vectors with n components. Then, Euclidean distance between objects i and j defined as

$$d(i, j) = \sqrt{(x1 - y1)^2 + (x2 - y2)^2 + (x3 - y3)^2 + \ldots + (xn - yn)^2)} \tag{3.7}$$

Another well-known measure is Manhattan distance or city block distance, named so because it is a distance determined in terms of the blocks between any two points in a city. It is defined by the following formula:

$$d(i,j) = |x1 - y1| + |x2 - y2| + \ldots + |xn - yn| \quad \textbf{(3.8)}$$

,where i, j are the data samples and x_i, y_j are the components of the corresponding data samples.

Minkowski distance is a generalization of Euclidean and Manhattan distances. It is defined as

$$d(i,j) = \sqrt[h]{|x1 - y1| + |x2 - y2| + |x3 - y3| + \ldots + |xn - yn|} \quad \textbf{(3.9)}$$

,where h is a real number such that $h \geq 1$.

3.3 Vector Space

The representation of set of documents in the same space as vectors is called Vector Space Model. It is based on the bag-of-words approach where the components of a vector are not ordered according to some rule. VSM was introduced by [47] and was used in the System for the Mechanical Analysis and Retrieval of Text (SMART) information retrieval system [46]. In VSM, the sentences involved in summarization

Table 3.1: Sentence X term matrix. 0-1 weighting scheme is used to fill the cells.

sentence	Turkey	Ankara	capital
sentence 1	1	1	1
sentence 2	1	1	0
sentence 4	1	0	1

are converted to vectors. Each vector component corresponds to a certain term in the document set. Term set is also called feature set. A feature set can consist of a single word, n-gram or a phrase. Each term corresponds to a single dimension in the vector space. A weight is assigned to a components of the vector and it shows the importance of the associated term. If the term appears in the text unit(sentence, phrase or other text parts) then the related component gets 1, otherwise it equals to 0. Also term frequency(tf) can be used to weigh a component. Term weighting schemes like $tf - idf$ might also be used to show the importance of the terms.

After each sentence is converted to a vector, the whole document set is represented by a term-sentence matrix A. It gives a suitable representation of textual units involved in processing. This matrix is an crucial point in many IR tasks. Document classification, document clustering, document scoring on a query etc. can be employed on it. To illustrate, let us consider the artificial data set given below.

1. The capital of Turkey is Ankara.
2. Ankara is situated in Turkey.
3. Ankara is one of the beautiful capitals in the world. The terms that occur in the

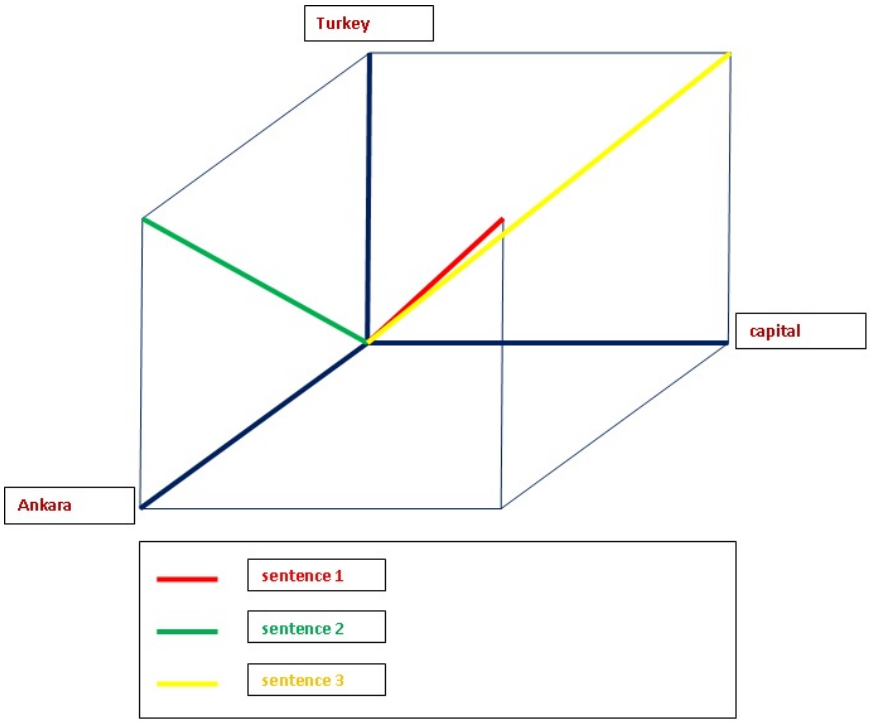

Figure 3.2: Sentence representation in Vector Space according to their terms.

set of sentences more than 2 times constitute the term set. Thus Ankara, Turkey and capital are the terms according to which the sentences are represented in the vector space and the term-sentence matrix A is created. The term-sentence matrix A is represented in Table 3.1 where the terms are organized in columns and the sentences are organized in rows. In addition, the sentences are shown in Figure 3.2 where each sentence is represented by its terms. Terms are the dimensions of the space and sentences are the vectors depicted in the space.

3.4 Latent Semantic Indexing

The main problem of search engines is synonymy and polysemy. Synonymy is a quality of a word group such that words in the group have equivalent meanings. This means that a group of words can have a similar sense. On the other hand, polysemy arises if a word has a multiple meanings. Depending on the context, the meaning of the word changes, but the lexical notation stays the same. Syntactic or semantic analysis needs to be done to determine the real sense of the word in the current situation.

Multiple meanings can be expressed in several ways. Every connotation is matched to one or more words in dictionaries. Thus these words form a group of words which are similar by their meaning. However, this approach does not help if such words are not detected in many text processing tasks.

On the other hand, a word might be used in different contexts. In each context the word plays different syntactic and semantic role. In some cases, the part-of-speech of the word is noun, in other instances the word is a verb. When the part-of-speech tag changes, the semantic functions of the word changes too. Consequently, words with several semantic functions is a main issue in the text processing.

Both synonymy and polysemy affect the accuracy of the search engines. If synonymous words are not linked, some relevant documents can not be retrieved. Only the documents which contain the exact matching words with the query are returned to the user. If a user searches "car" then documents containing "car" is found, but the documents with the word "auto" are not considered to be relevant. In another case, if the words with multiple meanings(polysemy) are not detected and not taken into account in indexing, then irrelevant documents might be returned as a searching result. All documents that contain the words in a query will be extracted from the data set without considering whether or not they satisfy the information need of a user. For example, if a user queries a search engine with a word "cat" then the system returns everything about "cat". The system does not distinguish whether "cat" is an animal or one of the utilities of Unix. All documents somehow related to the "cat" are included

Table 3.2: Sample data set from Deerwester et al. (1990).

documents	sentence
d1	**Human** machine **interface** for Lab ABC **computer** applications
d2	A **survey** of **user** opinion of **computer system response time**
d3	The **EPS user interface** management **system**
d4	**System** and **human system** engineering testing of **EPS**
d5	Relation of **user**-perceived **response time** to error measurement
d6	The generation of random, binary, unordered **trees**
d7	The intersection graph of paths in trees
d8	**Graph minors** IV: Widths of **trees** and well-quasi-ordering
d9	**Graph minors**: A **survey**

Table 3.3: Term-document matrix A for the sample data set.

terms	d1	d2	d3	d4	d5	d6	d7	d8	d9
computer	1	1	0	0	0	0	0	0	0
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
system	0	1	1	2	0	0	0	0	0
time	0	1	0	0	1	0	0	0	0
user	0	1	1	0	1	0	0	0	0
eps	0	0	1	1	0	0	0	0	0
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

in the result. Google returns for the query "cat" video about a cat, the website of the company which produces construction machines and engines, utility of Unix and also the website of World Cat Federation(queried 03.11.2013 17:50). Search engine could not determine the context of the word "cat". Thus, the meaning of the word should be defined exactly to achieve a good result in searches.

The context is an important component in differentiation of polysemy. It determines the co-occurrence patterns of the words. If a word is used in the economic context, then the word will be related to with economical terms. Even when the word does not have much sense in explaining the economical processes, it may have strong relations with the essential main words in the context. Consequently, the meaning of the word is defined by means of the words that occurs together in the same context. Different methods can be followed to overcome the problems of synonymy and polysemy.

Polysemy is known as word sense disambiguation and this is an open research area in NLP. Several methods like dictionary - knowledge based, supervised and unsupervised methods are used to solve the word sense ambiguity. Synonymy can be distinguished using WordNet. WordNet is a lexical dictionary for English. It classifies the words into synonym groups called synset. There is a definition for each word in a synset. The relations like hypernyms, hyponyms are defined on the basis of WordNet synsets. Hypernyms are abstract terms which are found in the higher levels of the WordNet lexical tree. Hyponyms are more specific terms which concretize the hypernyms. For instance, the hypernym "reference book" can be specified with hyponyms like "encyclopedia", "handbook" i.e. Also WordNet provides the polysemy count among the synsets. Polysemy count of the word is the number of synsets in which the word participated.

However, Latent Semantic Indexing brings a new solution to the above issues based on the word co-occurrence. In standard VSM, the terms are considered to be independent, thus their associations are not taken into account. By contrast, LSI discovers the relations between the words. It weights the words depending on the uncovered

Table 3.4: Cosine similarity of the terms in Vector Space Model.

terms	computer	human	interface	response	survey	system	time	user	eps	trees	graph	minors
computer	1.00	0.50	0.50	0.50	0.50	0.29	0.50	0.41	0.00	0.00	0.00	0.00
human	0.50	1.00	0.50	0.00	0.00	0.58	0.00	0.00	0.50	0.00	0.00	0.00
interface	0.50	0.50	1.00	0.00	0.00	0.29	0.00	0.41	0.50	0.00	0.00	0.00
response	0.50	0.00	0.00	1.00	0.50	0.29	1.00	0.82	0.00	0.00	0.00	0.00
survey	0.50	0.00	0.00	0.50	1.00	0.29	0.50	0.41	0.00	0.00	0.41	0.5
system	0.29	0.58	0.29	0.29	0.29	1.00	0.29	0.47	0.87	0.00	0.00	0.00
time	0.50	0.00	0.00	1.00	0.50	0.29	1.00	0.82	0.00	0.00	0.00	0.00
user	0.41	0.00	0.41	0.82	0.41	0.47	0.82	1.00	0.41	0.00	0.00	0.00
eps	0.00	0.50	0.50	0.00	0.00	0.87	0.00	0.41	1.00	0.00	0.00	0.00
trees	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.67	0.41
graph	0.00	0.00	0.00	0.00	0.41	0.00	0.00	0.00	0.00	0.67	1.00	0.82
minors	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.41	0.82	1.00

relations between them. In the following paragraphs the detailed explanation of LSI will be given.

Let us consider a simple example. We consider the document set shown in Table 3.2. The document set consists of 9 documents. Term-document matrix(A) shown in Table 3.3 can be derived from the document set. Each column represents a document, and each row corresponds to a term. Each cell of the matrix contains the weight of a term in a document. Here different weighting schemes can be used to fill each cell. In our example, term frequency is used.

The term-document matrix A can be used to determine different relationships like term-to-term or document-to-document similarities. If the cosine similarity measure is applied to the rows of the matrix, the similarities between the terms are established. In Table 3.4 term by term similarity matrix is shown. Similarity is calculated based on the word overlap. The larger the number of documents in which two terms are found, the greater the similarity between them. If two similar words are not in the

Table 3.5: Cosine similarity of the terms in Latent Semantic Space.

terms	computer	human	interface	response	survey	system	time	user	eps	trees	graph	minors
computer	1.00	0.21	0.85	0.99	0.85	0.30	0.99	0.99	0.25	-0.05	0.04	0.07
human	0.21	1.00	0.68	0.06	0.11	0.99	0.06	0.22	0.99	-0.01	-0.01	-0.02
interface	0.85	0.68	1.00	0.77	0.63	0.75	0.77	0.86	0.72	-0.15	-0.09	-0.06
response	0.99	0.06	0.77	1.00	0.83	0.16	1.00	0.99	0.11	-0.08	0.01	0.05
survey	0.85	0.11	0.63	0.83	1.00	0.20	0.83	0.82	0.15	0.48	0.56	0.59
system	0.30	0.99	0.75	0.16	0.2	1.00	0.16	0.31	0.99	0.005	0.003	0.003
time	0.99	0.06	0.77	1.00	0.83	0.16	1.00	0.99	0.11	-0.08	0.01	0.05
user	0.99	0.22	0.86	0.99	0.82	0.31	0.99	1.00	0.27	-0.09	-0.01	0.02
eps	0.25	0.99	0.72	0.11	0.15	0.99	0.11	0.27	1.00	-0.02	-0.02	-0.03
trees	-0.05	-0.01	-0.15	-0.08	0.48	0.005	-0.08	-0.09	-0.02	1.00	0.99	0.99
graph	0.04	-0.01	-0.09	0.01	0.56	0.00	0.01	-0.01	-0.02	0.99	1.00	0.99
minors	0.07	-0.02	-0.06	0.05	0.59	0.003	0.05	0.02	-0.03	0.99	0.99	1.0

same documents their similarity is low, however they might mean the same thing. In VSM, it is not important whether a word has synonym or has another meaning. The main point which determines the similarity of the terms is the number of documents in which they co-occur. In a similar way, document similarities can be defined by using column vectors instead of row vectors.

By contrast, LSI takes into account the patterns of co-occurrences of the words. It is supposed that the words used in similar contexts are related to each other and can share the similar meanings. If two words are similar to each other by their meaning then the group of the words used together with them do not differ considerably. This is because these words describe the same concepts. For example, if "user" is followed by "system" in text1 and "human" is used together with "system " in text2, then "user" and "human" are included in the chain human-system-user. The relation might be synonymy or polysemy or something else. But LSI infers the hidden relationships between the words. It weights the terms according to these relations. This case is shown in Table 3.5. It can be noticed that, the similarity between "human" and "user" is 0.22 in the LSI space, whereas the similarity between the same terms is 0 in the VSM space. In a similar way, many chains can be inferred from the given example: "human-interface-user", "human-computer-user" and so on. Such kind of chains are called second order co-occurrence in the literature [21]. These chains affect the similarity between the terms; the calculation takes into account the inferred chains. Consequently, the term-document matrix is created more accurately compared to simple word frequency based VSM.

SVD(Singular Value Decomposition) is used in LSA. SVD decomposes the real or complex valued matrix A into three matrices: U, V, Σ -unitary,orthogonal and diagonal matrices(Formula 3.10). The diagonal matrix has the singular values in the main diagonal. Hereinafter it is supposed that matrix A is real valued and it is decomposed into the three matrices.

Consider A_k the low rank approximation to A. Small perturbations to A_k correspond to the singular values in matrix Σ. This helps one to discriminate the significant impact of noise to the structure of A. Mathematically speaking, U consists of the left eigen-vectors and V consists of the right eigen-vectors of A. Σ is the diagonal matrix where singular values are made up its main diagonal.

$$A = U\Sigma V^T \tag{3.10}$$

,where U,V are orthogonal matrices and Σ is a diagonal matrix. Right and left singular vectors are calculated using AA^T and A^TA matrices respectively. The root values for the common eigen values corresponding to the left and right singular vectors are contained

in the main diagonal of Σ. These values are sorted in descending order. Corresponding eigen vectors of U and V are reordered to match their eigen values.

Eigen values represent the importance of the topics discussed in the documents. The noise in the document set can be eliminated if the sufficient number of the dimensions are selected in Σ. However, one needs to take care of while choosing the dimension k to represent the topics. If too large dimension is chosen noise might be included. Otherwise, if the small number of topics is selected it is likely that some essential topics will be discarded.

3.4.1 Semantic space of the documents

We get orthogonal matrices U, V and diagonal matrix Σ using SVD. Different semantic spaces can be created by using appropriate pairs of matrices. Semantic space of the terms can be created if the matrix U_k is multiplied by the matrix Σ_k. In a similar manner, the semantic space of the documents is created when the matrix V_k is multiplied by the matrix Σ_k. We are interested in the second space, since our method of summarization is based on the extraction of the sentences.

$$SpaceD = V_k \Sigma_k \tag{3.11}$$

Here $SpaceD$ is semantic space of the documents.
V_k is the representation of the documents in the reduced space with dimension r.
Σ_k is singular values of the matrix A which is enough to cover the main topics in the text. Documents are projected into this space by taking into account the main topics discussed in the collection. Multiplying the document matrix by Singular values gives us a document set where topics are rated according to their importance. However, documents in VSM are described by only expressing the existence of a word in the document only.

In multi-document summarization a term-sentence matrix is used to derive the different relationships by applying SVD to it. It is decomposed into U, V and Σ matrices. Eigen values are selected to represent the importance of the topics in the text. In the next step,

the semantic space is created using V and Σ matrices. Finally, Hierarchical Clustering algorithm is applied to the document set.

3.5 Hierarchical Clustering

Hierarchical clustering is one of the methods of clustering which builds a hierarchy of clusters. It does not need special settings of the parameters like the number of the clusters, the centroids of the clusters etc. Instead, it is required to set a measure and threshold of dissimilarity. At each level of hierarchy, the most dissimilar groups located at the lower level of hierarchy are merged to create a new cluster. Alternatively, the cluster at a top level can be split into two clusters which are placed at the next lower level of the hierarchy.

As supposed above, hierarchical clustering is classified as agglomerative and divisive. The former one is a "bottom-up" approach. Initially, each sample creates a cluster. In the succeeding steps, the pair of clusters with the smallest inter-cluster dissimilarity are merged to create a new cluster. A pair of cluster is merged at iteration and the iterations continue until a single cluster is left. This cluster is represented with the root node of the tree. Each inner node t corresponds to a cluster. The second hierarchical clustering is based on a "top-down" approach. In this case, the algorithm starts with the single cluster which contains all the samples. In the next steps, a cluster which satisfy a certain criterion(for instance, minimum squared error(MSE)) is recursively split into two new clusters. This procedure is repeated for as long as a cluster contains more than one sample remains. A cluster which contains a single sample is called singleton.

Both agglomerative and divisive clustering produce binary trees. Each node represents a cluster. The non-terminal nodes have two child nodes. In the divisive method, two child nodes are obtained when a parent node is split; whereas in agglomerative method, a new inner node is created when two child nodes are merged. The terminal nodes represent the singletons. Agglomerative algorithm starts with singletons. By contrast, divisive one starts with a single cluster that contains all of the samples.

The hierarchy of clusters can be shown with a dendrogram(Figure 3.3). The length of the lines that connect the clusters shows the value of dissimilarity between the

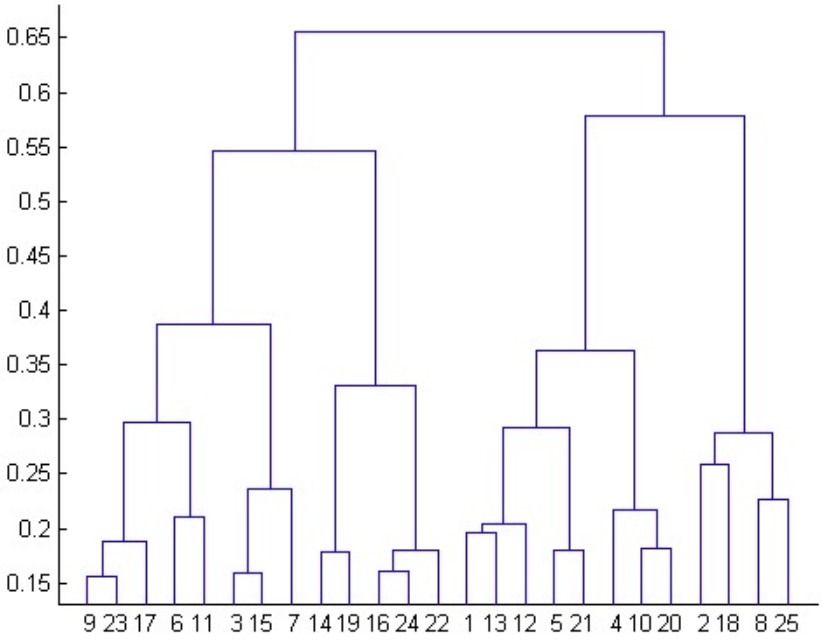

Figure 3.3: Dendrogram.

clusters. The height of the parent node depicted on the dendrogram is proportional to the magnitude of the dissimilarity between its two child clusters. The singletons are plotted at zero height. The magnitude of the dissimilarity monotonically increases as one moves to the higher levels of the tree.

3.5.1 Hierarchical agglomerative clustering

HAC(Hierarchical Agglomerative Clustering) algorithm begins with the singletons and it successively merges the clusters until the single cluster containing all samples is obtained. In each iteration, the most similar clusters are merged to form a new cluster. A similarity metric should be defined to determine the clusters to be merged.

Different strategies are followed to calculate the similarity/dissimilarity between the clusters which is called a linkage in literature. In single linkage the similarity between the clusters is determined by the most similar samples contained in distinct clusters. In complete linkage, the most dissimilar pair of samples included in different clusters, determines the similarity of the clusters. Group average linkage evaluates the cluster similarity based on all similarities including the inter- and intra- cluster similarities.

The main drawback of the single and complete linkage is that both of them are affected significantly by the noisy samples. Similarity between the clusters may change,

depending on the location of the noisy observations in the cluster. If the noisy sample is located far from other samples in the cluster then the distance is greater. It may cause erroneous clustering. However, group average linkage overcomes the stated problem. It uses all pairs of samples unlike single or complete linkage.

All in all, the similarity measures can be formulated as shown below:
single link: $d_{min(C_i,C_j)} = min_{p\in C_i, p'\in C_j}|p-p'|$
complete link: $d_{max(C_i,C_j)} = max_{p\in C_i, p'\in C_j}|p-p'|$
average linkage: $d_{avg(C_i,C_j)} = \frac{1}{n_i n_j}\sum_{p\in C_i}\sum_{p'\in C_j}|p-p'|$
where,$|p-p'|$ is the distance between two objects or points p and p', n_i is the number of objects in C_i.

3.5.2 Hierarchical divisive clustering

Initially, entire samples create a single cluster G. Next, the sample with the maximum average dissimilarity from the other samples in the cluster is chosen. This sample is included into cluster H which becomes a second cluster. At each following step, the sample in G, for which subtraction of the average inter-dissimilarity from the intra-dissimilarity is largest is chosen. In other words, if for a sample s the average distance(d) from the samples in H, minus the average distance(d′) from the other samples in G is maximum, then the sample s forms the next observation in the cluster H.

This procedure is repeated until the difference in averages becomes negative. That is, there is no sample in G similar or closer to the samples in H. As a result, the original cluster is divided into two clusters, the samples included into the cluster H. The obtained clusters are located at the second level of the hierarchy. The other levels are created by splitting an appropriate cluster from the top level of the hierarchy. The cluster with the largest diameter or the cluster with the largest dissimilarity among its members can be chosen to be split. This procedure of splitting the clusters is repeated recursively until all samples are in the singletons.

3.6 BFOS Algorithm

This algorithm was introduced by Breiman et al. [8] and extended by Chou et al. [10] Hereinafter this algorithm will be called the generalized BFOS algorithm and will be discussed in the context of multi-document summarization.

The generalized BFOS algorithm was used in classification and regression. The leaf nodes corresponds to the certain value or to the class. The main goal is to find an optimal classification or regression tree with minimum number of leaf nodes with the minimum squared error(MSE).

In the work of Chou et al., BFOS algorithm was extended and applied in many fields like Tree Structured Vector Quantization(TSVQ), variable order Markov Modeling etc. It was used to find an optimal pruned Tree Structured Vector Quantizer which enabled the coding with variable number of bits. The main parameters were rate and distortion.

3.6.1 Tree functionals

Let us assume that T is a tree. A tree consist of a root node, inner nodes and leaf nodes. A root node is placed on the top level of hierarchy and it is a starting point if one moves from a higher level to the lower levels of the hierarchy. Leaf node is terminal point of the tree which means that the node does not branch off. All other nodes between the root node and leaf nodes are named inner nodes. Every node contains a certain value and the pointers to the child nodes, to the parent node or to the neighbour nodes. A pointer is a physical or virtual non-duplicated address of the nodes. It can be the address of the node in physical memory or just the name of the node, but it has to be unique. A neighbour node is a node which is neither a parent node nor a child node. Thus, a leaf node can be defined as a node that do not point to any node except the parent node.

Any tree branched at any node of T is called a sub-tree of T and denoted as S. If S is rooted at any node of T except the root node of T and the leaf nodes contain the sub-set of the leaves of T($\tilde{T}$), then S is called a branch sub-tree of T. This type of sub-tree is designated as T_t. By contrast, if a sub-tree S is rooted at the root node of T, then the sub-tree S is named a pruned sub-tree of T and denoted as $S \preccurlyeq T$.

Functions defined on the tree or its sub-trees are called tree functionals. For instance, the number of nodes or the number of leaf nodes are tree functionals. Since each tree or their sub-trees correspond to the certain value. If the value of the functional is determined by the leaf nodes of the tree then the functional is linear. If it is defined by all nodes of the tree, it is affine. Functioanals can be also classified as increasing or decreasing. If the value of functional increases or decreases depending on the size of the tree then the functional is monotonic. The size of the tree equals the number of the nodes of the tree. If the functional increases monotonically as the tree grows, then the minimum value of the functional corresponds to the pruned sub-tree containing only the root node of the T.

3.7 Generalized BFOS Algorithm

As stated in Chou et al. the tree functionals($u1$ and $u2$) have to be defined correctly in order to use the generalized BFOS algorithm. In particular, $u1$ and $u2$ have to be increasing and decreasing functionals, respectively. These parameters are defined differently depending on the problem. For example, the number of leaf nodes can be $u2$ and the mean squared error function may be used as $u1$ in regression. If the number of the nodes are defined to be $u1$ and $u2$ equals the expected search time, then the generalized BFOS algorithm can be used in Tree-structured search tasks. Average length of the code can be $u1$ and the expected distortion can be $u2$ in Tree Structured Vector Quantization. In our case, $u1$ is defined to be the rate and $u2$ is the distortion. The definitions of the rate and distortion in the context of multi-document summarization are given in the following sections.

3.7.1 Euclidean space

The space which consists of all n-dimensional tuples $X = (e_1, e_2, e_3, \ldots, e_n)$ of real numbers is called Euclidean space(R^n). Any element of R^n is a point. Different operations like addition, multiplication by a scalar, finding the norm can be performed on the points in space R^n. If $x = (x_1, x_2, x_3, \ldots, x_n)$,$y = (y_1, y_2, y_3, \ldots, y_n)$ and $z = (z_1, z_2, z_3, \ldots, z_n)$ are points in space R^n, then $z = x + y$ is defined as

$$z_i = x_i + y_i \; for \; i = 1...n. \tag{3.12}$$

if $x = (x_1, x_2, x_3, \ldots, x_n)$ and α is a real number, then $z = \alpha x$ is defined as

$$z_i = x_i y_i \; for \; i = 1...n. \tag{3.13}$$

Let $x = (x1, x2, x3, \ldots, xm)$ be a set of points in R^n. The weighted sum of the given set of points x is called a convex combination

$$wx = \sum_{i=1}^{m} w_i * xi \tag{3.14}$$

if w is a weight vector and $w_1 + w_2 + w_3 + \ldots + w_m = 1, w_i \geq 0$. If x, y are in R_n then the convex combinations of x and y create a line segment.

$$\alpha x + (1 - \alpha) y \; with \; 0 \leq \alpha \leq 1 \tag{3.15}$$

If $x1$,$x2 \in C$ and $\alpha x1 + (1 - \alpha) x2 \in C$ then points of such line segments form a convex

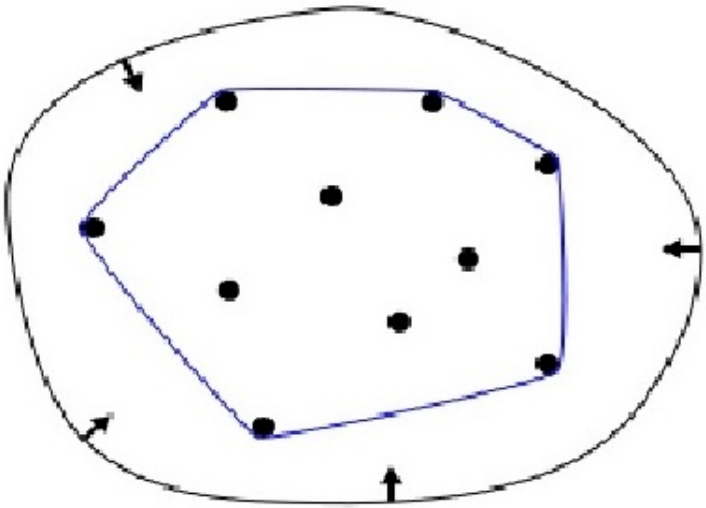

Figure 3.4: Convexity example.

set. For $C \subseteq R_d$, the set of all convex combinations of points in C is called the convex hull. It is a smallest convex set which includes C.

$$\sum_{i=1}^{|C|} \alpha_i x_i \tag{3.16}$$

where, $\forall i : \alpha_i >= 0$ and $\sum_{i=1}^{|C|} \alpha_i = 1$. It can be visualized using elastic band stretched around a set of points. The band touches the outer elements of X as shown in Figure 3.4. In a similar manner, the convex hull of the set X contains all boundary elements of X that is no one element falls outside of the convex hull.

3.7.2 Convexity of distortion-date functionals

Let $u1$(distortion) and $u2$(rate) be tree functionals defined on the tree T. In addition, let $u1$ be an increasing and $u2$ be a decreasing function. These functionals can be denoted as a vector u with two components $u1$ and $u2$. If a sub-tree S is pruned off and the corresponding $u(S)$ vector is calculated, then it is possible to calculate the effect of removing the sub-tree S. However, which functional to use for selecting the best

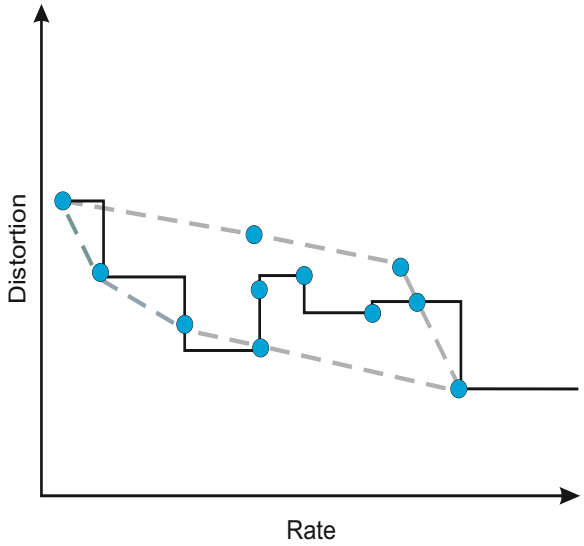

Figure 3.5: Distortion-rate graph and their convex hull, adapted from (Chou et al.(1989).

sub-tree is a fundamental issue. One can choose to use distortion as an optimization function. In this case, the most suitable sub-tree is the one which results in the minimal distortion. It turns out that, the minimal distortion is reached when any sub-tree of T is not pruned off.

Another choice may be the rate. A sub-tree S is eliminated if it gives the minimum value of the rate. The minimum rate is obtained if the sub-tree including only the root node is selected. Since the rate is a decreasing functional. However, this approach causes the distortion to reach the maximum value.

One of the solution lays in the convexity of distortion and rate(Figure 3.5). Points depicted on the distortion-rate plane form the convex sets. Since the line segments for each pair of points are located in the region bounded by the convex hull. Since distortion is an increasing and rate is a decreasing function, a pruned sub-tree having only the root node has the maximum value of $u1$ and the minimum value of $u2$; by

contrast, $u1$ reaches to its minimum and $u2$ its maximum value when the pruned sub-tree S contains all the nodes of the initial tree, T. Thus, these extreme points create the left upper corner and the right lower corner of the convex hull, respectively.

If one moves from the right lower corner to the left upper corner on the lower boundary of the convex hull, it is possible to locate the distortion-rate operating points which represent the best optimal pruned sub-trees of T. An optimal sub-tree trades off rate and distortion. It produces the minimum distortion for the given rate. It is sufficient to search an optimal sub-tree among the sub-trees represented by the distortion-rate points on the lower boundary of the convex hull. If $u(T), u(S_1), u(S_2), u(S_3), \ldots, u(S_n), u(t_0)$

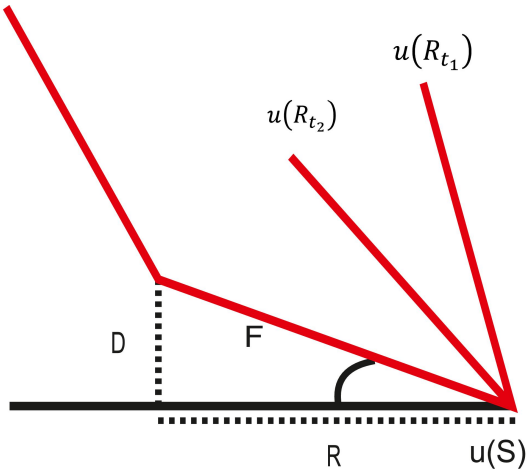

Figure 3.6: Distortion-rate ratio, adapted from (Chou et al.(1989).

are the distortion-rate points or the list of vertices clockwise around the convex hull then $t_0 \preccurlyeq Sn \preccurlyeq \ldots S_2 \preccurlyeq S_1 \preccurlyeq T$. The set of all optimal pruned sub-trees for the given values of the rate can be found using the operating points on the convex hull, because at least one branch of the nested sub-trees is located on the lower boundary of the convex hull. One can start with the full tree T and end up with the t_0 node(root node of T) passing through all operating points of the lower bound. Hence, the distortion-rate operating points located on the lower bound create the search space.

An optimal sub-tree S is determined by the magnitude of the slope (λ) of the face F. The optimal sub-tree S is located on the face F. Face is the step of the convex hull or the interval between the two subsequent operating points on the convex hull. All pruned sub-trees($R(t)$) of S are obtained by pruning off the single branch S_t from an interior node t(Figure 3.6). Hence, a vector $u(S)$ can be calculated using the following

formula:

$$u(S) = u(R(t)) + \Delta u(S_t) \tag{3.17}$$

,where $\Delta u(S_t)) = u(S_t) - u(t)$. In other words, $\Delta u(S_t)$ determines the change(increasing or decreasing) in the functionals. As shown in Figure 3.6, the slope corresponding to the vector $\Delta u(S_t)$ equals to $\Delta D/\Delta R$. The magnitude of the slope is not less than λ. If the magnitude of the slope was less than lambda one of the sub-trees of S would lie on the outside of the convex hull.

As shown in Chou et al., at least one sub-tree is located on the lower boundary of the convex hull. Thus, the sub-tree with the slope equal to the λ is selected as an optimal pruned sub-tree for the current tree S. Iteration of the pruning the most optimal sub-trees continues until the root node of T is reached or the certain criteria are met.

All in all, it is sufficient to find the inner node t with minimizes the magnitude of the slope. An optimal sub-tree is located on the face F. The slope of the optimal sub-tree is the same as the slope of F.

3.7.3 Implementation of the BFOS algorithm

Suppose that the initial tree is a binary tree T. Each node of the binary tree contains the following values:

λ- the magnitude of the slope

λ_{min} - the minimum λ value among the inner nodes of the current tree

$left(t)$- a pointer to the left child

$right(t)$- a pointer to the right child

$\Delta u(S_t)$- the amount of change when the branch S_t is pruned off

Initially, $\Delta u(S_t)$,λ_{min} values are calculated for each inner node of the tree T. The algorithm prunes the sub-trees until the root node remains.

Shortly, the generalized BFOS algorithm can be summarized in 3 steps.

1. Calculate the slopes of each sub-tree S_t rooted at inner node t.
2. Find the inner node t with the minimum λ value.
3. Prune the branch B_t rooted at t.

The tree pruning algorithm returns a pruned sub-tree of T that trades off distortion and rate.

3.7.4 Generalized BFOS algorithm in the multi-document summarization

As shown in the previous sections, the main two parameters or functionals, $u1$ and $u2$, have to be defined in order to use generalized BFOS algorithm. In our case, distortion and rate stand for the $u1$ and $u2$, respectively. As mentioned previously, distortion has to be an increasing functional and rate has to be a decreasing functional. If distortion expresses the information loss caused by the representing one or more sentence with another sentence and rate equals to the number of sentences then the monotonicity requirement of the functionals is satisfied.

Representative sentence contains the main topics discussed in the cluster of the sentences and it causes the minimal information loss. Centroid can be used as a representative sentence as described in Radev et al.. The words relevant to the main topics of the documents are included in the centroid. The relevancy is determined based on the statistics of the words.

On the other hand, a new sentence can be generated as a representative sentence. The sentence can be derived using the words in the cluster or it may contain the words from other sources. In addition, a sentence among the sentences included in the cluster can be selected as a representative sentence. A selected sentence covers the main content described in the cluster of sentences. The last approach is used in the current study. A sentence that causes the minimal distortion or information loss is chosen as the representative sentence.

3.7.5 Distortion-Rate framework

Distortion determines the information loss when the cluster of sentences is represented by a representative sentence. It is based on the distance metrics. Distortion contribution of each cluster(node) is defined as follows in the current context:

$$D = \sum_{s \in cluster} d(rs, s) \tag{3.18}$$

,where $d(.,.)$ measures the distance between a representative sentence(rs) and a sentence(s) in the cluster. By definition, the distortion contribution of each leaf node of the HAC tree equals to zero.

Another important parameter of the generalized BFOS algorithm is rate. Rate contribution as the number of sentences, words or symbols in the summary. In other words, it is amount of information given to the user about the topics in the document set in terms of sentences, words or symbols. Three approaches(sentence, word or symbol) are examined in the calculation of the rate in the current investigation.

4. IMPLEMENTATION

4.1 Sentence Parsing

Sentences are parsed according to the punctuations. Regular expressions are used to detect the end of the sentences. Alternatively, NLTK library can be used to parse the sentences. However, in this case, erroneous sentences are observed; where a whole sentence is divided into two or more sentences. Thus, the end of the sentences are detected by using regular expressions to avoid such cases.

4.2 Word Tokenization

Tokenization is the process of determining textual units or tokens that bear the semantic meaning. Generally tokens are considered to be words or terms and there is a little difference between the words and tokens. Tokens are the sequences of characters which are supposed to be a meaningful lexical unit. For instance, the contraction word "can't" can be considered as a single token "can't" or two distinct lexical units "can" and "t". It is up to the preprocessing task to split "can't" into the different lexical units or to use it as a single token. A word is a sequence of the characters which can be explained exactly and it has a certain syntactic role in the sentence. Thus, the token "t" is not a word, since it does not carry any meaning. Nevertheless, letter "t" is not considered as a word, it can be used as a token.

In the current work, the various uses of the apostrophe for contraction and possession are eliminated and the words are replaced with their complete form. For instance, are not is used instead of aren't and Google's is replaced with Google. In addition, shortened forms of the verb 'to be' in simple tense are extended; "I am" is used instead of "I'm" etc. Additionally, the numbers except the numbers that state the years are deleted. The last procedure improves the performance of LSI according to Johanna

Geiss(2011). After the appropriate words have been edited, NLTK word tokenization function is used to tokenize the sentences obtained from the previous step.

4.3 Normalization

Token normalization is the task of normalizing the tokens so that the tokens with the different character sequences are transformed to the same character representation. This procedure is applied in order to match the different tokens with the same meaning. To this end, several approaches are used. The most standard way is to create an equivalence cluster of the tokens as synsets in WordNet. For instance, anti-discriminatory and antidiscriminatory are included in the same equivalence cluster and the cluster is named after one of the members of the set. Alternatively, the tokens can be normalized according to the semantic relations. The most simple relation is the synonymy of the tokens or words. For example, automobile and car is located in the same group of tokens. When automobile or car is indexed they are represented with the same token. Additionally, case folding is performed for the normalization purposes. Case folding is the procedure of reducing all letters to lower case. It allows different instances of the word "automobile"("Automobile", "automobile", "AUTOMOBILE") to represent "automobile".

In our study, the last approach of normalization is applied to transform all tokens to the same representation form. Moreover, the dots are deleted from the dot separated tokens like abbreviations. The characters in an abbreviation are merged to create a single token. For instance, U.S.A is represented with USA. Normalization is crucial in our study, since it avoids the growth of the number of the terms.

4.4 Stemming

Words are used in different forms and the various suffixes are added depending on part of speech, person, mood etc. Despite, the form of a word changes, mostly the meaning of the word remains the same. For instance, the words organize, organizes, organizing are describing an action according to time and person, but the meaning of the verb is not changing. Additionally, there is a group of similar words which are derived by

Table 4.1: Porter stemmer. Rule groups.

Rule	Example
SSES -> SS	caresses -> caress
IES -> I	ponies -> poni
SS -> SS	caress -> caress
S ->	cats -> cat

using different suffixes. In many cases, it is useful to interpret these words as one word, since doing this reduces the number of words and it enables to detect or link semantically related words.

In order to carry out these tasks stemming can be used. Stemming reduces inflectional forms and derived forms of a word to a common base form. For example: "am", "are" and "is" matched to the base form "be"
"car", "cars", "car's", "cars'" are represented by the base form "car".

As can be noticed, stemming chops off the ends of the words and matches the given word to the completely different stem. Stem is the character sequence derived by stemming a word. Stem may or may not to be a meaningful word. For instance, stemming of the word "saw" returns "s" as a stem.

Porter's algorithm is the most common and widely used algorithm for stemming in English[40]. This algorithm is consisted of 5 stages. In each stage, a rule is selected among the set of rules according to the certain conventions such as deleting the suffix -ement or reducing -sses to ss.

In the first stage, a word is reduced according to the group of the rules shown on the table 4.1. In the next steps, the stemming algorithm continues depending on the number of syllables remaining in the word. For example, if m is located in a position greater than one then the suffix -ment is eliminated from a word.

4.5 Stop Word Elimination

Some words carry little importance and they can be excluded from the text without compromising the general idea. Mainly, such kind of words are used to connect words, sentences or to add additional meanings to words or sentences. Some of them do not have any meaning when they stay individually. For instance, the articles do not carry

any meaning when they are used as a single word. These words are called stop words in the literature.

Stop words are eliminated to avoid the dominance of the most frequent words over other words when similarity calculation is performed. Additionally, stop word elimination helps to reduce the number of the terms used to build term-sentence matrix. In the current project, standard list of the stop words are used.

4.6 Parts of Speech Tagging

Parts-of-speech tagging determines a class of a word or a lexical category of a word. The collection of tags used to label the words is named as a tag-set.

POS-tag of the word depends on the context where they are used. Some words may belong to several word categories. Thus, the POS-tag of a word can be determined by considering the words located around the given word. Hence, the entire sentence is taken into account in order to assign a tag to a word.

Parts-of-speech tagging is a kind of supervised learning problem, because it needs tagged corpora for learning purposes. The most popular and widely used corpuses are Brown corpus, Tree-bank and Conll2007.

The Brown corpus was collected from a variety sources using American English and it contains about a million words. It consists of 500 samples distributed over 15 genres equally. Texts included in the corpus were published in 1961.

4.6.1 Default tagger

The simplest approach of tagging is assigning the same tag to all words in the sentence. The most probable tag can be used as a default tag. In order to get the default tag, the frequency of a tag assigned to the words can be calculated and the most frequent tag can be used as a default tag. This type of tagging may be used in combination with other taggers. If a tag of the word is not determined after using the main tagger(for instance, Brill tagger) then the default tag is assigned to the given word.

4.6.2 Regular expression tagger

This tagger is based on the regular expressions and a tag is assigned to the given word according to the matched patterns. If a word ends with the suffix -ing then it can be assumed that the probable tag is present continuous form of a verb. In a similar manner, the following patterns can be derived according to the grammar rules:

1. ".*ing" -> "VBC"
2. ".*ed$" -> "VBD" simple past
3. ".*es$" -> "VBZ" third singular present etc.

4.6.3 The lookup tagger

Lookup tagger is based on the most frequent words in the corpus. At first, the most frequent words are determined and their most likely tags of the words are stored. Then, words are tagged according to the tags of the frequent words. If some of the words are not contained in the list of the frequent words they are assigned the tag "None".

4.6.4 Unigram tagging

Unigram taggers are built on the simple statistics of the tags for each token. The frequencies of the tags for each word are calculated. A tag with the highest frequency is assigned to the token. For instance, the word "frequent" is assigned the tag 'adjective', because in the training corpus it is more often used as an adjective. The tagger is trained by inspecting the tag of each token and storing the most likely tag for each token. When a token is encountered it is tagged with the stored tag.

4.6.5 General n-gram tagger

When a token is tagged with Unigram tagger, the context is not taken into account. A tag is assigned according to a prior probability of the tag. However, when N-gram tagger uses the tags of the N-1 preceding words to determine the tag of the current word. N-1 words play the role of the context where the current word occurs.

Table 4.2: Brill tagger output.

Phrase	to	increase	grants	to	states	for	vocational	rehabilitation
Unigram	TO	NN	NNS	TO	NNS	IN	JJ	NN
Rule1		VB						
Rule2				IN				
Output	TO	VB	NNS	IN	NNS	IN	JJ	NN
Gold	TO	VB	NNS	IN	NNS	IN	JJ	NN

4.6.6 Brill tagging

Brill tagging is one of the transformation based tagging methods and it was named after its inventor. In this method of tagging at first the tag of each word is guessed. Next, the mistakes are fixed and a correct tag is assigned to the word. This way, Brill tagger converts a bad tagging into a better one. This tagger is a supervised learning method, because it uses a tagged corpora to check whether a tag is assigned correctly or not. For instance, let us given the following sentence: "The President said he will ask Congress to increase grants to states for vocational rehabilitation".

Also let us assume that there are given the following rules are given:
a) replace NN with VB if the previous word is T0
b) replace TO with IN when the next tag is NNS

These type of rules are generated according to the following template: "replace T1 with T2 in the context C". T1,T2 and C are assigned a value and the numerous rules that use the variables in the template are created when the tagger is trained(Table 4.2).

4.6.7 Combination of the taggers

Several taggers can be combined to tag the words accurately. For example, a bigram, a unigram and a default tagger can be combined as follows:
1. The word is tagged with the bigram tagger.
2. If the bigram tagger is unable to assign a tag, the unigram tagger is used.
3. If after two steps the tag is not assigned to the word, the default tagger is executed.
The described above procedure is known as backoff. This is implemented by specifying one tagger as a parameter of another tagger as shown in the previous

example. Brill tagger and n-gram taggers are combined to assign a tag to a word in the current study.

4.7 Feature Set

Feature set consists of the terms(words, n-grams) considered to be important for the subsequent steps. In multi-document summarization, the important terms may be the terms related mostly to the main topics of the documents to be summarized. Different strategies are followed to measure the importance of the terms which are included in the feature set. One of them relies on the statistic of the words as tf or $tf-idf$, that is the terms with the frequency greater than the predetermined threshold are assumed to be important ones. Alternatively, pos-tagging can be applied as a feature creation procedure where the words with the appropriate pos-tags are considered to be suitable for the feature extraction. For the sake of simplicity, the terms with document frequency greater than 1 are included into the feature set in the current work. Also pos-tagging is applied to eliminate the words that do not carry much information(prepositions, articles etc.). Various cases are considered for choosing the pos-tags and the results are discussed in the section on experiments.

4.8 Term-Sentence Matrix Creation

Sentences are in the rows and terms are in the rows of the matrix. The cells are filled with $tf-idf$ value of the terms. This matrix is the main point of the algorithm, because the accuracy of the clustering depends on the values in the cells. Furthermore, the matrix is used as the input to LSI which weights the terms according to the relations of the terms or the concepts.

4.9 Latent Semantic Analysis

The created matrix in the previous step is given to the LSI module and it is decomposed into three matrices U, Σ and V. The sentences are projected to their semantic space to reflect the semantic relationships among the terms.

4.10 Clustering

The redundancy is one of the main issues in multi-document summarization. If the redundancy is not detected and eliminated, the sentences containing the same or the mostly similar information can be included in the summary. This is not desirable because the summary will contain the repeated content. Clustering is used in order to

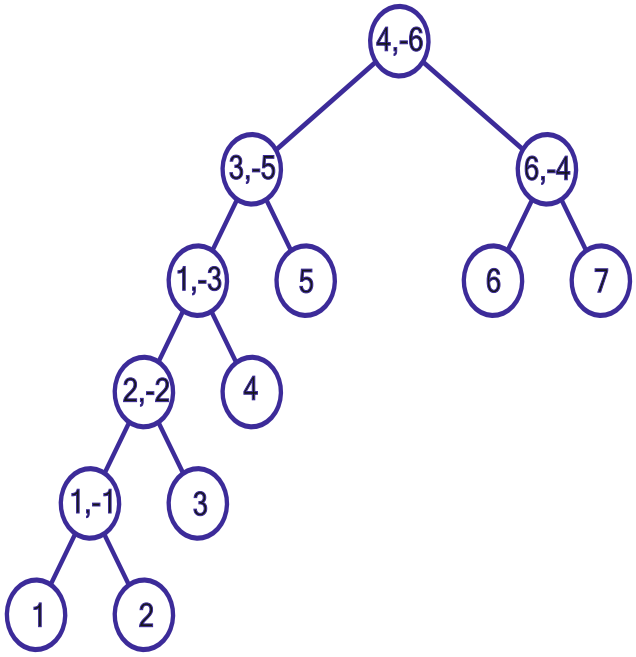

Figure 4.1: Clustering the sentences. A tree T is built using HAC(Hierarchical Agglomerative Clustering) algorithm.

overcome this issue. HAC algorithm is used to produce non-overlapping clusters and to build a tree which is pruned by the BFOS algorithm. The sentences are stored in the leafs of the tree and they create singletons. In the succeeding steps, the most similar clusters are merged. In each iteration, a new node appears in a higher level of the tree. Clustering continues until the tree is built or there is no more cluster to merge(Figure 4.1).

4.11 Tree Pruning

In the previous stage, clustering was used to detect the redundancy. In the current stage, BFOS algorithm is applied to eliminate the redundancy. BFOS algorithm prunes the tree and finds an optimal sub-tree which trades off between rate and distortion. Tree is pruned off until the certain criteria are satisfied. λ parameter, distortion or rate can

be used as a main criteria to stop the pruning algorithm. If an optimization parameter crosses the threshold, then the algorithm is finished. Summary is generated from the sentences in the clusters associated with the leaves of the tree. The sentence selection algorithm may be executed before the summary is created. One of the methods proposed by Murray [36], Steinberger [51] can be implemented to select the sentences to be included in the summary.

Initially, the sentences are stored in the leaf nodes of the tree. When an inner node t is pruned off, the representative sentence for the sub-tree S rooted at t is placed into inner node t. Since the sentences can be stored only in the leaf nodes of the tree, the inner node t is converted to a leaf node. In each iteration of the pruning algorithm, the

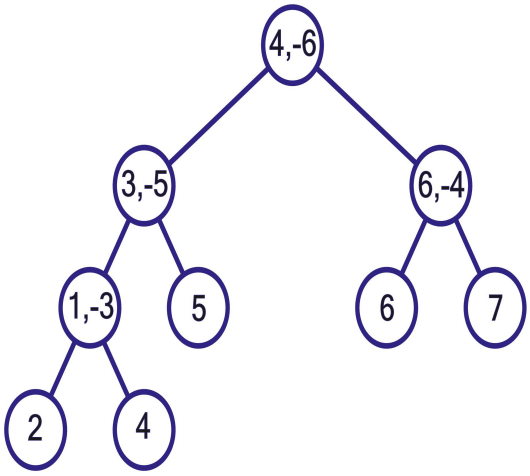

Figure 4.2: Pruning a tree T. A sub-tree S is pruned off.

sub-tree rooted at node with the minimum λ value is pruned. λ parameter determines the increase in distortion for the decrease in rate. The magnitude of λ is determined by the following formula:

$$\lambda = -\frac{\Delta D}{\Delta R} \tag{4.1}$$

where $\Delta D, \Delta R$ are the change in distortion and rate respectively if the node t is pruned off. ΔD is the difference between the total distortion before the sub-tree rooted at node t is pruned off(D_{prev}) and the total distortion after the sub-tree rooted at node t is cut off(D_{post}).

$$\Delta D = D_{prev} - D_{post} \tag{4.2}$$

The decrease in rate is calculated by the similar approach.

$$\Delta R = R_{prev} - R_{post} \tag{4.3}$$

where, R_{prev}, R_{post} are total rate before and after the pruning the sub-tree rooted at node t.

As can be noticed, λ is a slope of the pruned sub-tree $R(t)$ and $u(R(t))$ is located on the lower boundary of the convex hull. To illustrate, let us consider the tree T shown in Figure 4.2. As shown, the tree T consists of 7 sentences. The sentences are contained in the leaf nodes of the tree. For each sub-tree, a representative sentence is selected and it is stored in the root node of the sub-tree. For example, the sentence with id 2 is a representative sentence for the sub-tree S and it is contained in the inner node with id -2. In other words, the sentence with id 2 summarizes other sentences in the leaf nodes of sub-tree S(sentences with id 1 and 3). Hence, the sentence contained in the root of tree T forms the summary with one sentence length.

Let us suppose that sub-tree S is pruned off and rate is measured by the number of sentences. Since two sentences(1 and 3) are excluded and rate was equal to 7 before pruning(the number of sentences), the current rate equals to 5. If rate was measured in terms of the number of words or symbols, it would be equal to the number of the words or symbols contained in the sentences remaining after pruning. In a similar fashion, total distortion is updated. Sentences with id 1 and 3 are represented with a representative sentence. Hence, overall distortion is increased by the distortion caused by the pruning of the sub-tree S. The distortion values in the leaves of the tree T gives total distortion of the tree T. Overall distortion obtained before and after the pruning a sub-tree S is calculated as follows:

$$D_{prev}(S) = D(-1) + D(3->3) \tag{4.4}$$

$$D_{post}(S) = D(1->2) + D(2->2) + D(3->2) \tag{4.5}$$

,where D stands for distortion and "–>" means that a sentence is "represented by" another sentence. Since a sub-tree rooted at the node with id -1 has been pruned off before the pruning algorithm cuts off the node with id -2, $D(-1)$ is calculated as follows:

$$D(-1) = D(1->1) + D(2->1) \tag{4.6}$$

Actually, other leaves of the tree T ,not contained in the sub-tree S are involved in calculation of the corresponding distortions for the sub-tree S. But they are excluded

from the calculation, since they are not affected by the pruning.

$$D_{prev}(T) = D(-1) + D(3) + D(4) + D(5) + D(6) + D(7) \tag{4.7}$$

$$D_{post}(T) = D(-2) + D(4) + D(5) + D(6) + D(7) \tag{4.8}$$

The last 4 elements in the above formulas eliminate each other during the calculations of ΔD, since their values stay the same after pruning.

$$\Delta D(S) = D_{post}(T) - D_{prev}(T) = D(-2) - D(-1) - D(3) \tag{4.9}$$

The application of the generalized BFOS algorithm to the HAC tree can be recapped as follows. At the initial step, a representative sentence is selected for each inner node and λ is determined for each inner node. At each generic pruning step, the node with the minimum lambda value is identified. The sub-tree rooted at that node is pruned off. The root node of the sub-tree is converted to a leaf node. After each pruning step, the λ values of the ancestor nodes of this new leaf node are updated.

A summary of desired length can be created by selecting a threshold based on rate (the number of remaining sentences after pruning, the number of leaf nodes of the pruned tree). Another possibility for the choice of the stopping criterion may be based on the λ parameter whose magnitude monotonically increases with pruning iterations. When a large enough λ value is reached, it may be assumed that shortening the summary further eliminates informative sentences.

5. EVALUATION

The testing of the system has been performed on DUC-2002 [11] data set since the proposed system is designed to produce a generic summary without specified information need of users or predefined user profile. This data set contains 59 document sets. For each document set extraction based summaries with the length 200 and 400 words are provided. Document sets related to the single event are used for testing purposes.

Evaluation of the system is carried out using ROUGE package [32]. Rouge is a summary evaluation approach based on n-gram co-occurrence , longest common subsequence and skip bigram statistics [31]. The performance of the summarizing system is measured with Rouge-1 Recall, Rouge-1 Precision and F1 measure.

Evaluation of the system is performed under the following headings.

1. LSI vs. Vector Space
2. weighting schemes
3. POS-tagging
4. rate measures
5. distance metrics

5.1 LSI vs. Vector space

In the first evaluation scenario, the summarization is performed on the Vector Space and on the Latent Semantic Space. POS - tagging is applied and the feature set is formed using nouns, verbs, adjectives and adverbs. Terms are weighted using TF-IDF weighting scheme. Vector space is created after implementing all preprocessing stages as pos-tagging, feature set creation and weighting. The performance of the summarization system on the Semantic space and on the Vector Space is evaluated using Recall, Precision and F1 measure. As shown on the Table 5.1 the best results are

Table 5.1: Results. LSI vs. Vector Space

Space	Recall(R)	Precision(P)	F-measure(F1)
LSI	0.69	0.22	0.334
Vector Space	0.60	0.2	0.3

Table 5.2: Results. Weighting schemes

Weighting schemes	Recall(R)	Precision(P)	F-measure(F1)
Zero-one	0.62	0.20	0.30
Term frequency(TF)	0.64	0.20	0.31
TF-IDF	0.69	0.22	0.334

obtained when LSI is used, since LSI considers the relationships between the terms and weights the terms taking into account the co-occurrence statistics of the terms.

5.2 Weighting Schemes

Weighting schemes assign a weight to the terms depending on the existence in the sentence(zero-one weighting), frequency on the sentence(term frequency(tf)) or term statistics on the corpus(tf-idf). These approaches for the weighting is considered and tested. The results are shown on the Table 5.2. As expected, the best results are achieved using tf-idf weighting scheme. tf-idf combines local and global weighting schemes and thus takes into account the statistics of the terms in the sentence under consideration and the statistics of the terms in the entire sentence set. The lowest result gives zero-one weighting scheme as it does not consider the number of the occurrence or any other statistical features of the terms.

5.3 POS-Tagging

In this evaluation scenario several combinations of the POS-tags are considered. The following combinations of the tags are used in the feature set creation.

1. Noun, verb, adjective, adverb
2. Noun, verb, adjective
3. Noun, verb, adverb
4. Noun, verb The first feature creation scenario which is based on the main POS-tags gives the best result(Table 5.3). The POS-tags used in this scenario are the most

Table 5.3: Results. POS-tagging.

Pos-tag combinations	Recall(R)	Precision(P)	F-measure(F1)
N+V+Adj+Adv	0.69	0.22	0.334
N+V+Adj	0.63	0.19	0.29
N+V+Adv	0.66	0.21	0.31
N+V	0.68	0.219	0.333

Table 5.4: Results. Rate measures

Rate	Recall(R)	Precision(P)	F-measure(F1)
sentence	0.69	0.22	0.334
word	0.61	0.20	0.3
symbol	0.63	0.21	0.31

informative about the main topics of the document set. Consequently, the feature set contains the most informative and topic related terms.

Also the best result is obtained when noun and verb are used as the main tags for the features. This result does not differ considerably from the best result obtained using the first scenario. But Recall value decreases if adjectives or adverbs are added to the feature set(second and third row of the table).

5.4 Rate Measures

The number of the sentences as well as the number of the words and the symbols can be used as rate measure. The corresponding results for each rate measure are shown below on the Table 5.4. As can be seen, the best result is obtained when the number of sentences is used as the rate measure. Interesting result is obtained in other rate measures. The summarization performance is higher if the number of symbols is used as the rate measure rather than the number of words.

5.5 Distance Measures

As defined before, distortion is set to be the sum of the distance between the representative sentence and the candidate sentence. That is why the definition of the distance metrics affect the distortion. The distances metrics should be defined properly in order to get summaries that correlate well with manually extracted summaries. The results in terms of are shown below in Table 5.5. As can be noticed, the results for each

Table 5.5: Results. Distance measures

Distance	Recall(R)	Precision(P)	F-measure(F1)
Euclidean	0.6	0.48	0.54
Manhattan	0.548	0.49	0.517
Pearson	0.54	0.4	0.46
Cosine	0.55	0.41	0.47
Minkowski	0.58	0.49	0.53

Table 5.6: Candidate summary(produced by the proposed system) and 400E summary provided by DUC 2002 are compared with 200 word abstract created manually.

Summary	Rouge-1 Recall(R)	Rouge-1 Precision(P)	Rouge-1 F1
400E	0.31	0.55	0.38
candidate	0.30	0.57	0.39

distance measure differ considerably. F-measure equals to 0.54 for Euclidean distance; whereas the minimum result is obtained when Pearson distance measure is used to evaluate distortion. The difference between the maximum and the minimum results of the summarization system in terms of F-measure equals to 0.08. Consequently, the performance of the proposed summarization system depends on the selected distance metric. The performance of the system was compared with the summary provided by DUC-2002(Table 5.6). 400E stood for the extractive 400 word summary provided by DUC-2002 data set. It was created manually as an extractive summary for evaluation purposes. Candidate summary(CS) was produced by the proposed system. Both summaries were compared against a 200 word abstractive summary included in DUC-2002 data set. 200 word abstractive summary was considered as the model summary in ROUGE package. As shown, the summary of the proposed system gives better results in terms of Rouge-1 recall measure. However, the highest precision is achieved in the 400E summary. Generally, the proposed system outperforms the 400E summary, since F1-score, which takes into account precision and recall, is higher. In addition, the performance of the system was compared with the best systems [15], [16] of DUC-2002(Table 5.7). The results of the best systems(BEST) in terms of sentence recall and sentence precision are provided by DUC-2002. Sentence recall and sentence precision of the candidate summary(produced by the proposed system) were calculated by using 400 word extract based summary(provided by DUC-2002) and a candidate

Table 5.7: Candidate summary(produced by the proposed system) and 400E summary provided by DUC 2002 are compared with 200 word abstract created manually.

Summary	Sentence Recall(R)	Sentence Precision(P)
BEST	0.271	0.272
candidate	0.273	0.305

summary. Sentence recall and sentence precision are defined as follows:

$$sentence\ recall = \frac{M}{B} \tag{5.1}$$

$$sentence\ precision = \frac{M}{C} \tag{5.2}$$

where M is the number of the sentences included in both of the summaries(a candidate and 400 word summary provided by DUC-2002(400E)), C,B are the number of the sentences in the candidate summary and in a 400E summary, respectively. As shown,

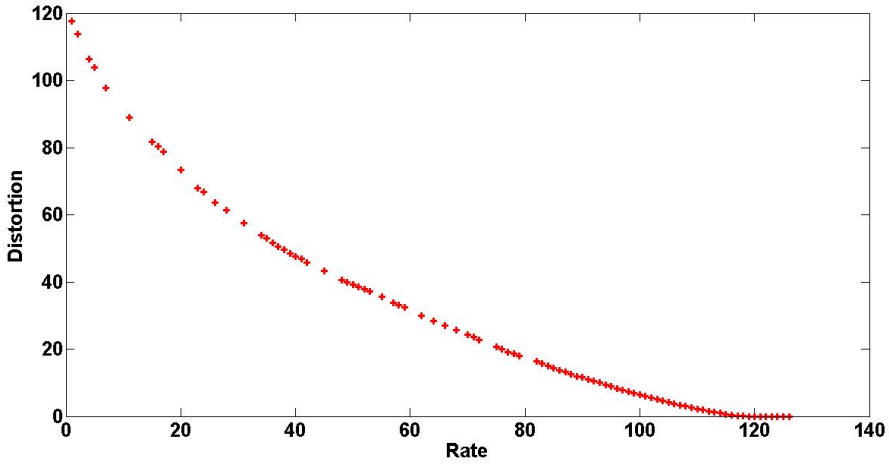

Figure 5.1: The relationship between distortion and rate. While rate is decreasing distortion is increasing.

the proposed system performs better than the best systems of DUC-2002 in terms of sentence recall. We are more interested in sentence recall because it states the ratio of the important sentences contained in the candidate summary if the sentences included in the 400E summary are supposed to be important ones. Furthermore, sentence precision is affected by the length of the candidate summary.

Summarizing the text can be considered as the compression of the text. Thus it is possible to depict the graph of dependence of distortion on rate (Figure 5.1). The

graph shows that as rate decreases distortion increases monotonically. Therefore, if distortion is assumed to be the information loss that occurs when the original text is summarized, then the summaries of different qualities can be produced by restricting rate (the number of sentences). Another graph shows the change of the λ value(Figure

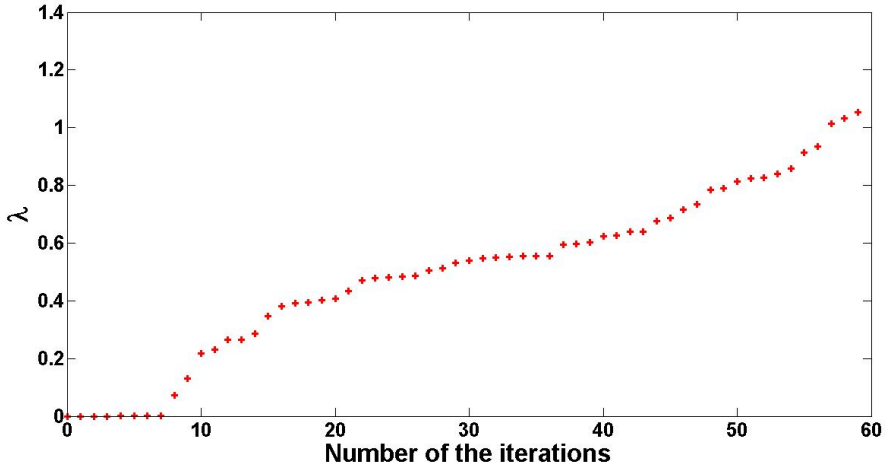

Figure 5.2: λ value of the pruned node. The change of λ value has upward tendency.

5.2). The pruning iteration number is on the X axis and λ value is on the Y axis. The λ value increases when the pruning iteration number increases. This indicates that the node with minimal λ value is selected in each iteration. Consequently, the sentences are eliminated so that increase in distortion is minimal for decrease in rate. All in all, the quantitative analyses show that the proposed system can be used as one of the redundancy reduction methods. However, in order to achieve the good results, the parameters of BFOS algorithm have to be set appropriately.

6. CONCLUSIONS AND RECOMMENDATIONS

The exponential growth of the electronic documents is a main obstacle in providing the users with the needed information. To overcome this problem, summarization techniques and approaches can be used. However, multi-document summarization brings other issue: redundancy elimination. This is the main task which has to be executed to avoid a repeated content. Different methods involving NLP, IR, statistics algorithms are developed to detect and eliminate the redundancy. Also the proposed system attempts to find a solution for the problem addressed.

In this investigation, the combination of the tree pruning algorithm and the clustering algorithm is explored. HAC algorithm is used to detect the redundancy in the text, whereas BFOS algorithm is applied to eliminate the redundant sentences. It is shown that if the parameters(distortion and rat) is set properly, generalized BFOS algorithm can be used to reduce the redundancy in the text.

The performance is evaluated with ROUGE package. The results suggest that the proposed system can perform better with additional improvements(combining with LSI etc.). Also it is stated that distance measure selection and noisy sentence inclusion have significance impact on the summarization procedure.

Future research will deal with the abstraction. A new sentence will be created(not extracted) when two clusters are merged. It will represent the cluster of sentences as well as summarize the other sentences in the same cluster.

REFERENCES

[1] **Aliguliyev, R.** (2006). A Novel Partitioning-Based Clustering Method and Generic Document Summarization, *In WI-IATW 06: Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, 626—629, Washington, DC, USA.

[2] **Arora, R. and Ravindran, B.** (2008). Latent Dirichlet Allocation Based Multi-Document Summarization, *In Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data (AND 2008)*, 91–97.

[3] **Barzilay, R. and Elhadad, M.** (1997). Using Lexical Chains for Text Summarization, *In Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization*, 10–17.

[4] **Barzilay, R.** (2003). Information fusion for multidocument summarization: Paraphrasing and generation, PhD thesis, DigitalCommons@Columbia.

[5] **Baxendale, P B.** (1958). Machine-made index for technical literature: an experiment, *IBM Journal of Research and Development*.

[6] **Bing, Q.**,**Ting, L.**,**Yu, Z.**,**Sheng, L.** (2005). Research on Multi-Document Summarization Based on Latent Semantic Indexing, *Journal of Harbin Institute of Technology*, **12(1)**,91—94.

[7] **Brandow, R.**, **Karl, M.**,**Olshen, R.A.**,**Lisa, F.R.** (1995).Automatic condensation of electronic publications by sentence selection, *Information Processing and Management: an International Journal - Special issue: summarizing text*, 675–685.

[8] **Breiman, L.**, **Friedman, J.H.**,**Olshen, R.A.**,**Stone, C.J.** (1984). Classification and Regression Trees, The Wadsworth Statistics/Probability Series, Belmont, CA: Wadsworth.

[9] **Chou, A. Philip**, **Tom Lookabaugh**,**Gray, M.** (1989). Optimal Pruning with Applications to Tree-Structured Source Coding and Modeling, *IEEE transactions on information theory*, **35(2)**.

[10] **Daniel, M.** (1997). From Discourse Structures to Text Summaries, *In Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, 82—88, Madrid, Spain.

[11] **DUC-2002**. (2002). Document Understanding Conference

[12] **Edmundson, H. P.** (1969). New methods in automatic extracting, *Journal of the ACM*, **16**, 264–285.

[13] **Gerard, S.**. (1988).Automatic text processing. Addison-Wesley Publishing Company.

[14] **Goldstein, J.**, **Mittal, V.**, **Carbonell, J.**, **Kantrowitz, M.** (2000). Multi-document summarization by sentence extraction, *In Proceedings of the ANLP/NAACL Workshop on Automatic Summarization*, 40–48.

[15] **H. van Halteren and Mittal, V.** (2002). Writing style recognition and sentence extraction., *In Proceedings of the workshop on automatic summarization*, 66—70.

[16] **Harabagiu, S.M. and Lacatusu, F.** (2002). Generating single and multi-document summaries with gistexter, *In Proceedings of the workshop on automatic summarization*, 30—38.

[17] **Hatzivassiloglou, V.**,**Klavans, J. L.**,**Holcombe, M.L.**,**Barzilay, R.**,**Kan, M.-Y.**, **McKeown, K. R.** (1999). Detecting text similarity over short passages: Exploring Linguistic Feature Combinations via Machine Learning, *In Proceedings of the 1999 Joint SIGDAT Conference on empirical Methods in Natural Language Processing and very large corpora*,203–212,College Park, MD, USA.

[18] **Hatzivassiloglou, V.**,**Klavans, J. L.**,**Holcombe, M.L.**,**Barzilay, R.**,**Kan, M.-Y.**, **McKeown, K. R.** (2001). SIMFINDER: A Flexible Clustering Tool for Summarization, *In NAACL Workshop on Automatic Summarization*,41–49. Pittsburgh, PA, USA.

[19] **Hahn, U. and Mani, I.** (2000). The challenges of automatic summarization, *IEEE Computer*, **33(11)**,29—36.

[20] **Hovy, E. and Lin, C.Y.** (1999). Automated Text Summarization in SUMMARIST, Mani,I. and Maybury, M.(eds.), Advances in Automatic Text Summarization, pages 81–94. The MIT Press.

[21] **Johanna Geiss**. (2011). Latent semantic sentence clustering for multi-document summarization, PhD thesis, Cambridge University.

[22] **Karen Sparck-Jones**. (1999). Automatic summarising: factors and directions, *In Advances in Automatic Text Summarization edited by Inderjeet Mani and Mark T. Maybury*, 1—12, MIT Press, Cambridge MA, USA.

[23] **Kathleen McKeown**, **Judith Klavans**, **Vasilis Hatzivassiloglou**, **Regina Barzilay**, **Eleazar Eskin**. (1999). Towards Multidocument Summarization by Reformulation: Progress and Prospects, *In Proceedings of AAAI*, Orlando, Florida.

[24] **Kenji Ono**, **Kazuo Sumita**, **Seiji Miike** (1994). Abstract generation based on rhetorical structure extraction *In Proceedings of the 15th International Conference on Computational Linguistics*,**1**, 344—384, Kyoto, Japan.

[25] **Kupiec, J.**, **Pedersen J.,**, **Francine, Ch.** (1995). A Trainable Document Summarizer, *SIGIR '95 Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 68–73.

[26] **Kolda, T. and Dianne, P.** (1998). A semidiscrete matrix decomposition for latent semantic indexing information retrieval, *ACM Transactions on Information Systems (TOIS)*,**16(1998)**, 322-346.

[27] **Kenji Ono**, **Kazuo Sumita**, **Seiji Miike**. (1994). Abstract generation based on rhetorical structure extraction *In Proceedings of the 15th International Conference on Computational Linguistics*,**1**, 344—384, Kyoto, Japan.

[28] **Karen Sparck-Jones**. (1999). Automatic summarising: factors and directions, *In Advances in Automatic Text Summarization*, 1-–12, MIT Press, Cambridge MA, USA.

[29] **Landauer, T.K.**, **Foltz, P.W.**, **Laham, D.** (1998). Introduction to Latent Semantic Analysis, *Discourse Processes*, **25**,259—284.

[30] **Lee, D. and Seung, S.** (1999). Learning the parts of objects by non-negative matrix factorization, 401(1999), 788–791.

[31] **Lin, C.Y. and Hovy, E.** (1999). Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics, *In North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLTNAACL- 2003)*,71–78.

[32] **Lin, C.Y.** (2004). Rouge: A package for automatic evaluation of summaries, *In Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004*.

[33] **Lin, C.Y. and Hovy, E.** (2002). From Single to Multi-document Summarization: A Prototype System and its Evaluation, *In Proceedings of the ACL conferenc*,457—464, Philadelphia, PA, USA.

[34] **Luhn, H.P.** (1958). The Automatic Creation of Literature Abstracts, *IBM Journal of Research Development*, **2(2)**,159–165.

[35] **Marcu, D.** (1997). From Discourse Structures to Text Summaries, *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, 82-88.

[36] **Murray, G.**, **Renals, S.**, **Carletta, J.** (2005). Extractive summarization of meeting recordings, *In Proceedings of the 9th European Conference on Speech Communication and Technology*.

[37] **Morris A.H.**,**George, M.K**, Dennis, A.A. (1992). The Effects and Limitations of Automatic Text Condensing on Reading Comprehension Performance, *Information Systems Research*, **3(1992)**,17-35.

[38] **McKeown, K. R.**, **Klavans, J. L.**, **Hatzivassiloglou, V.**, **Barzilay, R.**, **Eskin, E.** (1999). Towards multidocument summarization by reformulation: Progress and prospects, *In Proceedings of AAAI-99*,453—460, Orlando, FL, USA.

[39] **Ono, K.**, **Kazuo' S.**, **Seiji' M.** (1994). Abstract Generation Based on Rhetorical Structure Extraction, *COLING '94 Proceedings of the 15th conference on Computational linguistics*, 344-348.

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

[41] **Radev, D. R.**, **Jing, H.**, **Budzikowska, M.** (2000). Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation and user studies, *In ANLP/NAACL Workshop on Summarization*, 21–29,Morristown, NJ, USA.

[42] **Radev, D. R.**, **Blair–goldensohn, S.**, **Zhang, Z.** (2001). Experiments in Single and Multi-Docuemtn Summarization using MEAD, *InFirst Document Under- standing Conference*, New Orleans,LA.

[43] **Radev, D. R.**, **Jing, H.**, **Stys, M.**,**Tam, D.** (2004). Centroid-based summarization of multiple documents, *Information Processing and Management,*, **40**,919-938.

[44] **Radev, D. R.**, **Teufel, S.**, **Saggion, H.**,**Lam, W.**, **Blitzer, J.**, **Celebi, A.**,**Liu, D.**, **Drabek, E.** (2003). Evaluation Challenges in Large-scale Document Summarization, *In Proceeding of the 41st meeting of the Association for Computational Linguistics*, 375—382, Sapporo, Japan.

[45] **Deerwester, S.**, **Dumais, T. Susan**, **Landauer, Thomas K.**, **Richard Harshman**. (1990). Indexing by latent semantic analysis, *Journal of the American Society of Information Science*, **41(6)**,391–407.

[46] **Salton, G.** (1971). The SMART Retrieval System Experiments in Automatic Document Processing, Prentice Hall, Englewood Cliffs, NJ, USA.

[47] **Salton, G.** (1979). Mathematics and information retrieval, *Journal of Documentation*, **35(1)**, 1—29.

[48] **Seno, E. and Nunes, M.** (2008). Some experiments on clustering similar sentences of texts in portuguese, *JProceedings of the 8th international conference on Computational Processing of the Portuguese Language*.

[49] **Steinberger, J. and Jezek, K.** (2004). Text Summarization and Singular Value Decomposition, *In Proceedings of ADVIS'04*,Springer Verlag.

[50] **Steinberger, J. and Jezek, K.** (2004). Using Latent Semantic Analysis in Text Summarization and Summary Evaluation, *Proceedings of ISIM '04*,93–100.

[51] **Steinberger, J. and Jezek, K.** (2007). Text Summarization within the LSA Framework, PhD Thesis.

[52] **Teufel**, **Simone**, **Marc Moens** (1997). Sentence extraction as a classification task, *ACL/EACL workshop on Intelligent and scalable Text summarization*, 58–65.

CURRICULUM VITAE

Name Surname: Ulukbek Attokurov

Place and Date of Birth: Kyrgyzstan, 27.04.1986

E-Mail: uluk86@mail.ru

B.Sc.: Kyrgyz-Turkish Manas University

M.Sc.: Istanbul Technical University

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- Ulukbek Attokurov and Ulug Bayazit. "Multi-document summarization using distortion-rate ratio", *In Proceedings of the ACL Student Workshop 2014*, pages 64-70, June 22-27 2014, Baltimore, Marryland USA.