Islamic University, Gaza, Palestine
Research and Postgraduate Affairs
Faculty of Engineering
Computer Engineering Department

Multi-document Arabic Text Summarization

Karim S. AL Harazin

Supervisor
Dr. Wesam M. Z. Ashour

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

April 2015

DEDICATION

To My Mother, Father,

To my wife,

To my Sister,

To My brothers,

To My Friends,

ACKNOWLEDGEMENTS

Praise is to Allah, the Almighty for having led me at every point of my biography. And best pray and peace on Prophet Mohammed.

I would also like to take this chance to thank my research supervisor, ***Dr. Wesam M. Z. Ashour****, for guiding and helping me throughout this research and other courses.*

As well I would like to thank the discussion committee, ***Dr. Mohammed Alhangouri*** *and* ***Dr. Ihab Zakout****, for their guides and commentaries.*

ملخص الدراسة

مع النمو السريع للبيانات على شبكة الانترنت، اصبح هناك حاجة ملحة لأنظمة التلخيص الالي للنصوص. الكثير من الابحاث اجريت على اللغة الانجليزية وغيرها من اللغات الاوروبية. على الرغم من التطور في الابحاث، الا أنه لايوجد دعم كبير للغة العربية.

هذا البحث يعنى بإنشاء نظام تلخيص آلي للنصوص المتعددة المصادر المكتوبة باللغة العربية. إن اكبر التحديات التي تواجه تلخيص النصوص متعددة المصادر هي: ازالة البيانات الزائدة عن الحاجة و اعادة ترتيب الجمل بشكل مقروء. في هذا البحث أيضا، ناقشنا امكانية استخدام طرق التلخيص الآلي للنصوص من مصدر واحد لتلخيص النصوص متعددة المصادر. بالاضافة الى ذلك قمنا بعرض نظام تلخيص الّي يعتمد على نظرية البنية البلاغية للمستندات CST. استخدام ال CST يساعد على تحديد العلاقات البلاغية بين الجمل من مختلف المصادر.

إن الاختلاف الرئيسي بين النظام المقترح وباقي الأنظمة العربية المستخدمة للتلخيص الآلي هو استخدام البنية البلاغية للمستندات في استخراج الاجزاء المهمة من النصوص.

في عملية تقييم أداء النظام، تم استخدام معيار ROUGE الذي يقيس مدى تشابه نتيجة التلخيص الآلي بالتلخيص اليدوي. تم مقارنة النظام المقترح بعشرة انظمة اخرى، النظام المقترح الذي يعتمد على البنية البلاغية للمستندات كان من بين أول ثلاثة أنظمة أعطت نتائج جيدة.

الكلمات المفتاحية: التلخيص الآلي للنصوص، تلخيص النصوص متعددة المصادر، استخلاص الخصائص، نظرية البنية البلاغية للمستندات، معالجة اللغة الطبيعية

ABSTRACT

With the rapid growth of data on the internet, there is an essential need for automatic summarization systems. A lot of automatic text summarization researches have been done for English and other languages. Recently, there has been growing interest in the Arabic language by researchers. Many of these researches concerned with single document Arabic text summarization. Multi-document summarization facing many challenges, the main challenges are: redundancy removal, and sentence reordering.

In this research, we discussed the possibility of using a single document summarization methods for multi-document summarization, also we proposed a system for multi-document Arabic text summarization based on cross document structure theory. The CST based method help to identify the semantic relationships between sentences across different documents. For redundancy removal we create a novel approach based on splitting the similar sentences into smaller units to eliminate unnecessary ones, and realign the rest of units to form a non-redundant sentence.

For evaluation, a Recall-Oriented Understudy for Gisting Evaluation ROUGE evaluation measure is used. The proposed system is applied on news domain using TAC 2011 MultiLing Pilot dataset. The proposed system is compared by ten peer summaries provided from the dataset. The evaluation results show a good performance for CST based method compared to the other peer systems summaries.

***Keywords**: Automatic Text Summarization, Multi-document summarization, Feature Extraction, Natural Language Processing, Cross-document structure theory*

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AATSS	Automatic Arabic Text Summarization System
ANLP	Arabic Natural Language Processing
CR	Compression Rate
CST	Cross document Structure Theory
DF	Document Frequency
DT	Determiner
DUC	Document Universal Conferences
DUC2002	Document Understanding Conference 2002
ER	Entity Recognition
FB_MDS	Feature Based Multi-document summarization
GP	Genetic Programming
GSM	General Statistic Method
HMM	Hidden Markov Model
IDF	Inverse Document Frequency
IN	conjunction, subordinating or preposition
IR	Information Retrieval
JJ	Adjective
MDS	Multi-document summarization
NLP	Natural Language Processing
NN	noun, singular or mass
NP	noun phrase
P	Precision
PDA	Personal Digital Application

POS	Part of speech
PP	prepositional phrase
R	Recall
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
RST	Rhetorical Structure Theory
TF	Term Frequency
TF-DF	Term Frequency × Document Frequency
TF-IDF	Term Frequency × Inverse Document Frequency
VBN	verb, past participle
VP	verb phrase
VSM	Vector Space Model

LIST OF TABLES

LIST OF FIGURES

Chapter 1

1. Introduction

1.1. Topic Area

With the rise of the Internet users, the amount of information available on the Web is increasing rapidly. By the end of 2013 the number of the Internet users reached 2.7 billion user [1].

The need for systems which can automatically summaries documents is becoming ever more desirable. For this reason, text summarization has quickly grown into a major research area. Text summarization can be helpful in many fields such as medical area, legal area, news area and any other fields because it saves time and resources.

Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document. As the problem of information overload has grown, and as the quantity of data has increased, so has interest in automatic summarization. Technologies that can make a coherent summary take into account variables such as length, writing style and syntax. An example of the use of summarization technology is search engines such as Google [2].

Generally, there are two approaches to automatic summarization: extraction and abstraction. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. In contrast, abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate. Such a summary might contain words not explicitly present in the original. Research into abstractive methods is an increasingly important and active research area, however due to complexity constraints, research to date has focused primarily on extractive methods [2].

1.2. Categories of text summarization

There are often several related views which can be used to characterize text summarization. The main categories used to classify summarization are mentioned as follow [3].

1.2.1. Number of documents

- Single document summarization: tries to summarize one document and extract informative sentences only from it

- Multi-document summarization: tries to generate one summary from multi-documents by extracting the most important sentences from each document and put them in readable order.

1.2.2. Number of languages in the document

- Mono language summarization: deals with generation of summary from document written in one language.

- Multi-languages summarization: unlike the mono language in this type the documents contains at least two different languages.

1.2.3. Form of summary

- Abstractive summarization: it is the hardest task for computer researchers to solve this type successfully as it is concerned with semantic and language complexity. The summary containing sequence of words not present in the original document.

- Extractive summarization: it consists of words, sentences and paragraphs that are completely appear in the original document. This approach suffers from inconsistencies, lack of balance and cohesion. Also some sentences may be extracted out of the context.

1.2.4. Query/generic summary

- Generic Summarization: Provides an overall summary of all the information contained in a document. Answers the question "what is this document about?”

- Query-Relevant Summarization: Specific to information retrieval applications, for example the snippets below each result returned by a search engine. Attempts to summarize the information a document contain pertaining to specific search terms. Answers the question "what does this document say about?”

1.2.5. Level of processing

- Surface-level approaches,: represents information in notions of shallow features and their combination. Shallow features include e.g. statistically salient terms, positionally salient terms, terms from cue phrases, domain-specific or a user’s query terms. Results have the form of extracts.

- Deeper-level approaches: produce extracts or abstracts. Abstracts summaries uses synthesis involving natural language generation. They need some semantic analysis e.g. can use entity approaches and build a representation of text entities (text units) and their relationships to determine salient parts. Relationships of entities include thesaural relations, syntactic relations, meaning relations and others. They can as well use discourse approaches and model the text structure on the base of e.g. hypertext markup or rhetorical structure.

1.2.6. Purpose of the summary

- Indicative summaries: give abbreviated information on the main topics of a document. They should preserve its most important passages and often are used as the end part of information retrieval (IR) systems, being returned by search system instead of full document. Their aim should be to help a user to decide whether the original document is worth reading. The typical lengths of indicative summaries range between 5 till 10% of the complete text.

- Informative summaries: provide a substitute (“surrogate”, “digest”) for full document, retaining important details, while reducing information volume. Informative summary is typically 20-30 % of the original text.

- Critical or Evaluative summaries: capture the point of view of the summary author on a given subject. Reviews are typical example, but they are little bit out of scope of nowadays automatic summarizers.

1.3. Difference between single document and multi-document summarization

Single document and multi-document summarization are similar to each other, both of them try to summarize a given text regardless of the source of that text single or multiple documents.

Summarizing multi-documents is more difficult than summarizing single document, even with a very large document. This difficulty arises from inevitable thematic diversity within a large set of documents. A good summarization technology aims to combine the main themes with completeness, readability, and conciseness.

There are two major challenges in multi-document summarization which are redundancy elimination and sentence ordering.

- Redundancy elimination: having a set of related documents may result in redundant and duplicated information that must be eliminated from the final summary. Redundancy eliminations one of the main differences between single and multi-document summarization. In case of single document, the chance of having duplicated sentences is very low. Given multiple documents, however, the information stored in different, documents inevitably overlaps with each other. Hence, effective methods that merge information stored in different documents and if possible, contrast their differences are highly desired in the case of multi-document summarization [4].

- Sentence ordering: A summary with improperly ordered sentences confuses the reader and degrades the quality/reliability of the summary itself. Sentence

ordering for multi-document summarization is a hard process [5,6]. the main reason is that unlike single document, multi-document do not provide a natural order of a text to be the basis of sentence ordering judgment [7].

1.4. Arabic Natural Language Processing

1.4.1. Arabic language

The Arabic language is the largest living member of the family of Semitic languages in terms of speakers. It is closely related to Amharic and Aramaic. Arabic is today spoken by more than 200 million people in the Arab World, and it is an official language in 22 countries. With the growth of the Arab internet users, the Arabic text has also grown.

Arabic language has three forms; Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectal Arabic (DA). CA, MSA, and DA forms include classical historical liturgical text, news media and formal speech, and predominantly spoken vernaculars and have no written standards, respectively

1.4.2. Challenges in Arabic NLP

Semantic processing of Arabic language considered to be much more complex than other languages like English and European languages. This complexity comes from the nature of the Arabic language which is highly derivational. In Arabic language one word could have up to seven synonyms, for example the following word 'معركة' could have more than ten synonyms as shown in Table 1.1

Table 1.1: example of word synonyms

Word	معركة
Synonyms	اشتباك, خلاف, شجار, صراع, عراك, عمل عسكري, عملية, عملية عسكرية, قتال, معركة, معركة ضارية, موقعة, نزاع

As shown in Table 1.1, the word may have large number of synonyms. The replacement of any of these synonyms depends on the context. For example suppose the two examples in Table 1.2, the first column hold the original sentence and the second column hold the modified sentence after replacing the word ‘معركة’ by its synonym. The replacement of

the word 'شِجار' in first sentence will not make the sentence semantically correct because the synonym 'شِجار' is related to few number of persons, not to a war between two armies. In contrast the synonym 'شِجار' at the second sentence will lead to the same meaning as the original sentence.

Table 1.2: example of synonyms replacement

Original Sentence	Replaced Sentence
دارت المعركة بين الجيشين اليوم صباحاً	دار الشجار بين الجيشين اليوم صباحاً
حصل عراك بالأيدي بين الفريقين	حصل شجار بالأيدي بين الفريقين

Any Arabic NLP tool should take care of this challenge. There are some aspects that slowed down the progress in Arabic NLP compared to the accomplishments in English and other European languages [8, 9], which include the following.

- The absence of capitalization in Arabic, makes it hard to identify proper nouns, titles, acronyms, and abbreviations.
- Arabic is highly inflectional and derivational, which makes morphological analysis a very complex task.
- Diacritics (vowels) are, most of the time, omitted from the Arabic text, which makes it hard to infer the word's meaning and therefore, it requires complex morphological rules to tokenize and parse the text.
- Shortage of Arabic corpora, lexicons and machine-readable dictionaries.

1.5. Document representation

Document representation is important for text summarization process. one of the most widely used document representation is Vector Space Model (VSM). The VSM is widely used in text mining, information filtering, text clustering, information retrieval, and text summarization. VSM represents a document as a vector of weighted terms, the number of vector dimensions equals the number of distinct terms or phrases appear in the document. Vector space model also known as "bag of words" model [10]. If a term occurs in the document, its value in the vector is non-zero. Several different ways of computing these

values, also known as (term) weights, have been developed. The VSM is represented as a matrix as the following:

$$\begin{array}{c} \\ D_1 \\ \vdots \\ D_n \end{array} \begin{array}{c} \begin{array}{ccc} T_1 & \cdots & T_l \end{array} \\ \begin{pmatrix} w_{11} & \cdots & w_{tl} \\ \vdots & \ddots & \vdots \\ w_{1n} & \cdots & w_{tn} \end{pmatrix} \end{array} \qquad 1.1$$

Where *t, n* is the number of terms and documents respectively, in the document collection. Columns represents all terms in the document collection and rows represents the number of documents in the collection. w_{11} is the term weight of the term 1 in document 1. w_{tn} is the weight of term *t* in the document *n.* One of the best known schemes is term frequency- inverse document frequency (TF-IDF) weighting[11].

The TF-IDF measure the term local and global significance to the document called. The TF is used to reflect the local importance of a term within the document, if the terms occur at least one time the value of TF will be 1+log(TF) otherwise TF value will be zero. IDF measure the global importance of the term across all document. IDF value derived by dividing the total number of documents in the collection by the term's frequency of occurrence within that chosen document and taking its log. If the term is so common in all documents in the collection, then its IDF value will equal zero. The final weight of the term is computed by multiplying TF and IDF, this weight is then normalized by the square root of the square of the sum of the TF*IDF for all unique terms in the document collection.

Because of its ease of use and efficiency, the VSM considered as a standard text representation method. Many representation methods implemented based on VSM method.

1.6. Research question

There are a lot of methods could be used for automatic text summarization. Due to the importance of summarization problem, we need to answer the following questions.

1. What is the better method could be used for Arabic multi-document text summarization?

2. How could we adapt a single document summarization method to be used for multi-document summarization?
3. What is the best features to be used for scoring sentences?
4. How could we identify the semantically related sentences from different documents talking about the same topic?
5. How to solve the problem of data redundancy?
6. How to measure the system performance?

1.7. Thesis contribution

- More support of Arabic multi-document summarization.
- Two methods for Arabic multi-document summarization are developed. The first method is based on feature extraction while the second is based on cross document structure theory CST
- Incorporate semantics in the process of text summarization.
- An Arabic annotated CST dataset is created by translating an existing English dataset.
- A new method is implemented for redundancy removal based on sentence splitting and merging.

1.8. Thesis Organization

The rest of this thesis is organized as follows: Chapter 2 introduces the related works that are relevant to our work. Chapter 3 describes the proposed methodology and approaches, in chapter 4 we will show the results of our work, and finally Chapter 5 presents the conclusions and future works

Chapter 2

2. Related Works

The main challenge in text summarization process is the extraction of the most informative. The text summarization recently has more attention by researchers. In this chapter we will look at some of early and recent single and multi-document summarization works.

In general text summarization is the process of summarizing single or multi-documents by extracting the most important information. The summarizer must be sure that the final summary is ordered in a good manner to be readable. Automatic text summarization uses computer tools and algorithms in order to produce the summary. Works on automatic text summarization started more than 50 years ago. Works on Arabic text summarization is still new research field. Early works on Arabic text summarization started less than 10 years ago.

The works on automatic text summarization could be categorized into many of categories. Next we will review three main categories of them which are feature based methods, clustering based methods, and graph based methods.

2.1. Feature based methods

One of the widely used methods in automatic text summarization is based on features extraction methods. Text features could be used to reflect the importance of the sentences. Here are some of features that have been considered for sentence selection.

2.1.1. Term frequency

Term frequency or term weight is a common feature that measure the importance of a term in the document collection. The most common measure of term weight is TF-IDF.

2.1.2. Similarity with title

Sentence similarity with document title measure how the sentence is relevant to the document. The more similar the sentence the more important.

2.1.3. Sentence length

Very short sentences are in general not included in the summary since they hold less information. Also very long sentences are not included in the summary. Long sentences could be compressed in order to extract the informative part of it.

2.1.4. Sentence position

The sentences located at the beginning of the document are considered to contain the most important information.

2.1.5. Named entity

Sentences containing a named entity like a person, organization, and place are considered important to the document.

After computing the features, the sentence represented as a vector of normalized features scores. The total sentence score is computed as the aggregate features scores. Next the sentences ordered according to their aggregate scores from heights to lowest. Summary generation then is done by selecting the top sentences in the list until reaching the desired summary length.

The early works on feature based summarization was started by Luhn [12] in 1985. He proposed a system for generating abstract of scientific papers. In order to determine the important sentences to be included in the final summary two features was used: word occurrences and sentence relative position. After computing all sentences scores the system then specifies sentences with high scores to be included in the abstract.

Baxandale [13] used sentence location as a sentence selection method. The sentences located at the beginning or at the end of paragraph is considered to be important and is included in the final summary.

Later researches added new features for sentence selection. Edmundson [14] added two new features in addition to word frequency and sentence location. The new features are pragmatic words: cue words, title and heading words. Cue words such as “significant”, “key idea”, and “hardly”. Baxendale compared his work against manual extracts; a score of 44% was the result of his experiment.

Feature based automatic text summarization could be improved using feature selection (FS) process. FS process identifies the most important features that can represent the data. The reason behind using FS techniques include reducing dimensionality, removing irrelevant and redundant features, reducing the amount of data needed for learning, improving algorithms predictive accuracy increasing the constructed models comprehensibility, and improves the quality of system results [15, 16].

Binwahlan et al. [17] created a novel text summarization model based on swarm intelligence known as Practical Swarm Optimization (PSO). They use PSO as a training model for features weights. The following equation is used for sentence scoring

$$Score(S) = \sum_{i=1}^{s} w_i \times score\_f_i(s) \qquad 2.1$$

Where, $Score(S)$ is the score of the sentence S, w_i is the weight of the feature *i* produced by PSO, i = 1-5 showing that 5 text features where used and $score\_f_i(s)$ is the score of the feature *i*.

Abuobieda et al. [18] used genetic concept as an optimized trainable features selection mechanism. The Document Understanding Conference (DUC2002) used to train their proposed model. Firstly they make a basic preprocessing on the input document using porter stemmer. In their model they used five text features which are title-feature, sentence position, thematic word, sentence length, and numerical data feature.

Kwaik [19] proposed a model to automatically summarize Arabic text using text extraction. Various steps are involved in the approach: preprocessing text, extract set of features from sentences, classify sentence based on scoring method, ranking sentences and finally generate an extract summary. The main difference between their proposed system and other Arabic summarization systems are the consideration of semantics, entity objects such as names and places, and similarity factors in her proposed system. The system performs several stages to achieve text summarization: data acquisition, pre-processing and feature extraction, scoring, and ranking and generating summary. she compare her system with Shakr system. The results show that her system is better. The

first method of our thesis is based on [19]. However, we will adapt her system for multi-document summarization and improve the system by adding the support of Arabic lexical database WordNet [20] for similarity measurement and term weighting. WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. WordNet can thus be seen as a combination of dictionary and thesaurus.

Sobh [21] introduced an Arabic extractive text summarization system. This system integrates Bayesian and Genetic Programming (GP) classification methods in an optimized way to extract the summary sentences. The system is trainable and use manually labeled corpus. They extract features for each sentence based on Arabic morphological analysis and part of speech tags in addition to simple position and counting methods. After extraction, they use Bayesian and GP in different manners to generate some versions of the summary either by integrating the two results or by selecting the max score between them. Using GP method didn‘t add any powerful value to the model as the result said. Using Bayesian alone increase the precision of the summary and saving the time needed for GP computation. They have four type of summarization system according to the combination between Bayesian and GP which are: (i) Bayesian, (ii) GP, (iii) Bayesian and GP, (iv) Bayesian or GP. From evolution they found that using Bayesian or GP achieves the highest F-measure between the four approaches which reach to 0.599 when they used only five features (sentence length, sentence paragraph position, sentence similarity, number of infinitives, and number of verbs).

2.2. Clustering based methods

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other clusters [22]. Documents are usually written such that they address different subjects or topics. Many researchers incorporate the idea of clustering in automatic text summarization. Before perform clustering methods on

sentences, the starting point is to extract unique words list from document set treating these words as features. Sentences represented using eq. (**1.1**):

Using clustering methods, the sentences which are highly similar to each other are grouped into one cluster. The result of clustering then will be a number of clusters of similar sentences. Usually the cosine similarity measure is used to measure the similarity between two sentences. Once sentences are clustered, sentence selection is performed by selecting sentence from each cluster. Sentence selection is then based on the closeness of the sentences to the top ranking TF-IDF in that cluster. Those selected sentences are then put together to form the final summary.

Judith D. Schlesinger et al [23] used CLASSY (Clustering, Linguistics, And Statistics for Summarization Yield) to generate single or multi-document (cluster) summaries of Machine Translation (MT) translated documents. The proposed method used trimming rules to shrink sentences, identify sentences and organize the chosen sentences for the final summary. Here the thought was to design a multi-lingual summarization technique. CLASSY structural design made up of five steps: preparation of raw texts, trimming of sentences, scoring, redundancy elimination and sentence organizing. This method can also be used for machine translated edition of Arabic document as well as English document. The trimming method is truly dependent on language and the quality of summarization very much depends on the translation quality of machine.

Radev et al. [24] proposed a system called it MEAD. MEAD is a centroid-based method to score sentences based on sentence-level and inter-sentence features, including cluster centroids, position, TF-IDF (term frequency inverse document frequency). Their method analyzes the cluster centroids and generates a pseudo-document that includes the sentences with the highest TF-IDF term values. Then, the system give each sentence a score based on the sentence similarity with the centroids, the sentence position within the document, and the sentence length.

Sarkar [25] presented a multi-document text summarization system, which clusters sentences using a similarity histogram based sentence-clustering algorithm to identify multiple sub-topics (themes) from the input set of related documents and selects the

representative sentences from the appropriate clusters to form the summary. The system consists of three component:

- Similarity histogram based incremental sentence clustering method that groups similar sentences by keeping each cluster at a high degree of coherency.
- Cluster ordering scheme that orders the clusters in decreasing order based on the relevance or information richness of the clusters.
- Representative sentence selection scheme that selects one sentence from each cluster.

They compare their system with other five peer systems, results show that, their system has performance compared to the baseline summary.

2.3. Graph based methods

Recent researches make use of graph theory in automatic text summarization. A graph is an ordered pair G = (V, E) comprising a set V of vertices or nodes together with a set E of edges or lines, which are 2-element subsets of V (i.e., an edge is related with two vertices, and the relation is represented as an unordered pair of the vertices with respect to the particular edge). In the context of the text summarization, the vertices are represented by sentences and edges are the weight between two sentences. All the graph based methods have the same concept in text summarization, all of them includes: data preprocessing, construction of graph model, link analysis and ranking algorithm, and finally summary generation.

Rada et al. [26] introduced the TextRank graph-based ranking model for graphs extracted from natural language texts. Their graph-based ranking algorithm consists of the following main steps:

1- Identify text units that best define the task at hand, and add them as vertices in the graph.

2- Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or un weighted.
3- Iterate the graph-based ranking algorithm until convergence.
4- Sort vertices based on their final score. Use the values attached to each vertex for ranking/selection decisions.

Zhang et al. [27] proposed a graph based summarization system based on the hub-authority framework. They integrated surface features into a graph-based ranking system, three features were included which are first sentence, cue phrase and sentence length. Firstly using sentence clustering, the system detect the sub-topics in multi-documents and extract the feature words (or phrase) of different sub-topics. Secondly, all feature words and the cue phrase are used as the vertex of Hub and all sentences are regarded as the vertex of Authority. If the sentence contains the words in Hub, there is an edge between the Hub word and the Authority sentence. Finally the summary is generated according to the sentence ranking score of all sentences.

Thakkar et al. [28] presented a new method based on TextRank graph-based summarization algorithm. Their idea take into account the flow in the text, since the extracted parts, usually sentences, are taken from different parts of the original text. This can for instance lead to very sudden topic shifts. The idea behind their method of extracting sentences that form a path where each sentence is similar to the previous one is that the resulting summaries hopefully have better flow. This method uses shortest path algorithm for summary generation. The summary is created by taking the shortest path that starts with the first sentence of the original text and ends with the last sentence. Since the original text also starts and ends in these positions, this will hopefully give a smooth but shorter set of sentences between these two points rather than taking top ranking sentence like in Text Rank.

2.4. Cross document Structure Theory (CST) based summarization

The idea of cross-document structural relationship is to investigate the existence of inter-document rhetorical relationships. These rhetorical relations are based on the CST model (Cross-document Structure Theory). Usually, the documents that discuss the same topic contain semantic relations between their sentences, these relations call CST relations. Examples of semantic relations are “Identity”, “Overlap”, “Description”, “Subsumption”, and “Historical background”. Full descriptions of the CST relations are given in [29]. Many researches used CST to extract the most important sentences for multi-document summarization. In the work presented by Zhang et al. [30], they replace low-salience sentences with sentences that maximize total number of CST relations in the final summary. To determine the CST relations between the sentences, they conducted experiments, in which human subjects were asked to find these relations over a multi-document news cluster.

In the research by Ibrahim almahy et al. [31], they used Cross-document structure theory with the incorporation of some thread properties to produce an extractive summary for the thread conversation. They developed sentence scoring based on model selection technique. They selected sentences relying on their number of relations, then the result was improved by using model selection optimization technique to improve the process of sentence selecting and consequently improving the summary result. The model selection technique can assign different weights for each relation, and it searches for the optimum weights set that produce a concise summary.

Researches [32, 33] used CST in multi-document summarization though analyzing the relations between the sentences and then select the important ones, either by choose the sentences with high number of relations [33], or by selecting sentences depend on their type of relations [32].

All of the above mentioned works and others, only applied on English, Brazilian Portuguese texts and Japanese texts [34, 35]. Until the writing of this thesis there are no works deal with Arabic language. For this purpose, we have created an Arabic CST dataset by translating an existing English annotated CST relations dataset, the translation

done by a human translator. Our experiments show that the CST based summarization has a better evaluation results compared to other methods.

2.5. Final remarks

It is notable that each of the above mentioned methods has its own advantages towards multi-document summarization. However there are some issues and limitation. The feature based method is knowledge poor in term of capturing contextual information contents that exist in the sentences and multiple documents. These limitations are due to the sentence scoring process which depends only on flat feature representation of a sentence while omitting cross-document relationships between text units in different documents. Clustering based methods are also have an issues. In clustering based method, sentences are ranked according to the similarity with cluster centroid which represents frequent occurring terms. Thus, this method is also considered to be knowledge poor in term of its inability to capture contextual information contents that exist in the sentences. Finally the approach to graph based methods have resulted in positive feedback from the multi-document summarization research. The resulting graph is also able to capture distinct topics from unconnected sub-graphs. However since this approach depends heavily on sentence similarity to generate graph, it only treats sentence as bag of words without "understanding" the text. This would produces the final summary to be not complete enough specifically for an informative summary generation [36].

Chapter 3

3. Multi-document Arabic automatic text summarization

This chapter explains the methodology which followed in this research. Various stages are performed to achieve text summarization. One of the main steps in Arabic text summarization is the pre-processing of input text.

3.1. Pre-processing step:

Before generating a summary, the documents are needed to run under a number of pre-processing operations. Preprocessing phase aims to transform the collected Arabic text documents into an easily accessible representation of texts that is suitable for the text summarization. The pre-processing in general includes the following operations (Figure 3.1): tokenization process, normalization, stop-words removal, stemming, and document indexing.

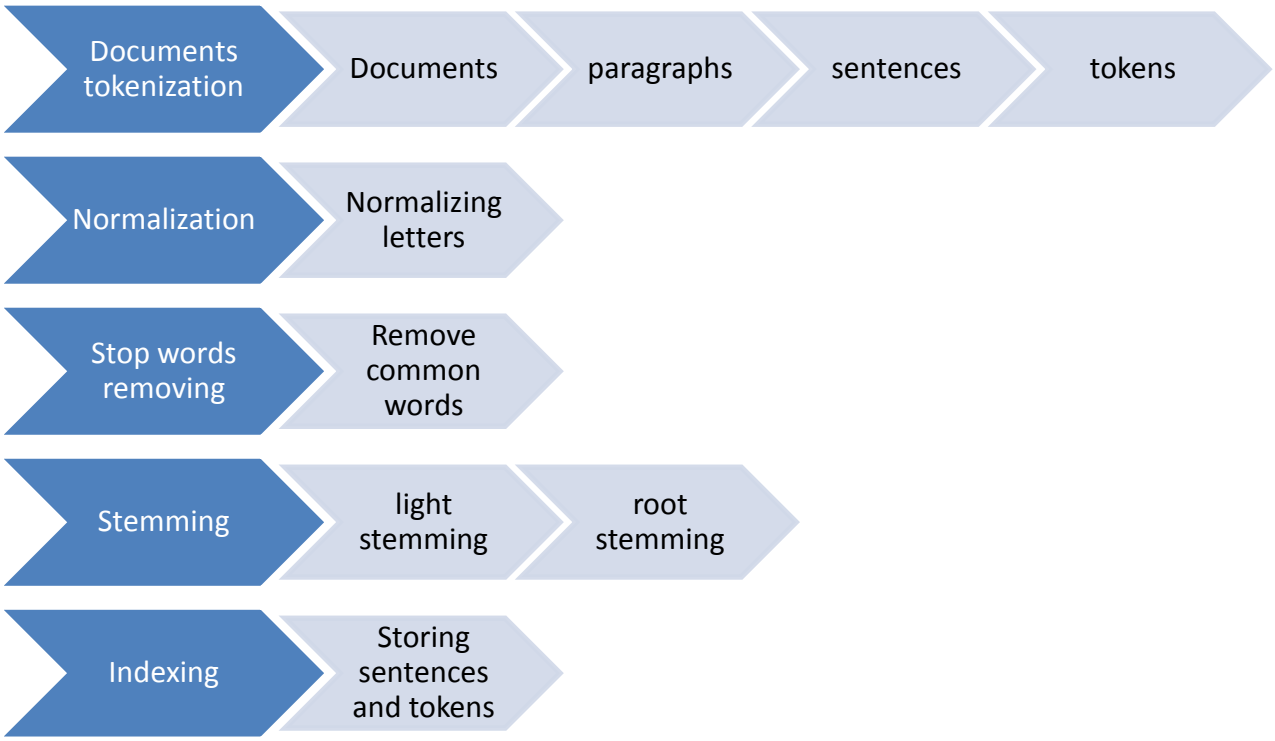

Figure 3.1: Arabic text pre-processing steps

3.1.1. Tokenization

For any input text the pre-processing starts with tokenization, the process of splitting the documents into its final units. These units could be characters, words, numbers, sentences or any other appropriate unit [37]. For less complex languages, tokenization usually involves splitting punctuation, and some affixes off of the words. On the other hand, morphologically rich languages, like Arabic, require a more extensive tokenization process to separate different types of clitics and particles from the word [38]. However, Tokenization closely related to the morphological analysis. The tokenize process is responsible for defining word boundaries such as white spaces and punctuation marks, multiword expressions, abbreviations and numbers[39].

In this research, the simplest form of tokenization is used, which only splits off punctuation and non-alphanumeric characters from words. For example the sentence in Table 3.1, is separated into 14 tokens, note that the tokenization process tokenize the dot as a token, but this will be eliminated in normalization step.

Table 3.1: Example of text tokenization

طائرة الرحلة 447 بعثت 24 رسالة خطأ قبل وقت قصير من أن تختفي.													
طائرة	الرحلة	447	بعثت	24	رسالة	خطأ	قبل	وقت	قصير	من	أن	تختفي	.

For text summarization the input documents is tokenized into sentences and each sentence into tokens (Algorithm 3.1).

Algorithm 3.1: Tokenization Algorithm

input: Document set

output: List of word tokens

foreach Document in Document set **do**
    Split document into sentences
    foreach sentence in document **do**
        Split sentence into tokens
    end
end

3.1.2. Normalization

Normalization is the process of unification of different forms of the same letter. Before stemming and stop word removal, corpus was normalized as follows:

- Remove punctuation.
- Remove diacritics (primarily weak vowels).
- Remove non letters.
- Replace أ, إ, and آ with ا.
- Replace ي with ى.
- Replace ة with ه.

3.1.2.1. Remove Diacritics

In Arabic language there are special notations called diacritics. It used for Arabic grammar and difference in the meaning according to word position in sentence and its Part of Speech (POS). Table 3.2 shows the diacritics which are removed from the text

Table 3.2: Arabic diacritics

~ً ٌ ٍ َ ُ ِ ّ ْ

3.1.2.2. Remove Punctuations

As any language, punctuations are used to organize the text and give the sentence a powerful meaning. The punctuation in the text summary does not have any value, so we remove all punctuations which are not full stop. Table 3.3 shows punctuations that should be removed when appear in the text.

.C= + - _)"، º >< | \ ¿ ' ! @ # $ % ^ & *) ~ ø × ÷ ': .¡ ,

Table 3.3: Punctuations list

3.1.2.3. Letters Normalization

Alef letter can be written in many forms such as آ ,إ ,أ, sometimes the word comes with different Alef styles. Alef letter can appear in the input text in three forms which are

"أ", "إ", and "ا". Converting Alef letter to 'ا' style will help in feature extraction, term weighting processing. The same normalization process is applied on the 'ي' and 'ة' which converted into 'ى' and 'ه' respectively.

3.1.3. Stop words removing

Stop words are frequently occurring, insignificant words that appear in an article or web page (i.e. pronouns, prepositions, conjunctions, etc.). Words like (هذه, تكون, قد, بين, أين, كان, على, انه) are considered stop words, which are non-informative. Stop words list are removed because they do not help determining document topic and to reduce features [40]. There is no definite list of stop words which all Natural language processing (NLP) tools incorporate. Not all NLP tools use a stop list. Some tools specifically avoid using them to support phrase searching. There are several techniques to remove the stop-words from the text. In our research, the stop list is stored in a hash map and removed from the documents to be summarized [41].

3.1.4. Stemming

Stemming is the term used in linguistic morphology and information retrieval to describe the process for reducing inflected (or sometimes derived) words to their word stem, base or root form—generally a written word form. The stem needs not to be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Algorithms for stemming have been studied in computer science since the 1960s. Many search engines treat words with the same stem as synonyms as a kind of query expansion, a process called conflation [42].

There are two major approaches that are followed for Arabic stemming. One approach is called light stemming (also called stem-based stemming) by which a word's prefixes and suffixes are removed; the other one called Root-based stemming (also called aggressive stemming) which reduces a word to its root. Another two approaches that have been researched are Statistical stemming and manual constructing of dictionaries; the last one is inefficient and therefore not so popular. Studies show that light stemming outperforms aggressive stemming and other stemming types [43]. The following section discusses the two approaches.

3.1.4.1. Root stemmer

The root is the basic form of word from which many derivations can be obtained by attaching certain affixes, so we produce many nouns and verbs and adjectives from the same root [44]. A root based stemmer main goal is to extract the basic form for any given word by performing morphological analysis for the word [45]. **Table 3.4** shows an example of root 'كتب' along with some of its derivations that can be obtained from that root.

Table 3.4: Example of word root and its derived stems

Root	Derived Stems
كتب	كتاب
	كاتب
	كتب
	مكتبة

The problem in this stemming technique is that many different word forms are derived from an identical root, and so the root extraction stemmer creates invalid conflation classes that result in an ambiguous query which leads to a poor performance [46]

3.1.4.2. Light stemmer

Arabic light stemming is the process of removing a small set of prefixes and suffixes with no attempts to remove infixes or returning the word's root [47]. Many light stemming methods like Leah [48] stemmer classifies the affixes to four kinds of affixes: antefixes, prefixes, suffixes and postfixes that can be attached to words. Thus an Arabic word can have a more complicated form, if all these affixes are attached to its root [49, 50]. Table 3.5 shows an example of a word and its affixes

Table 3.5: A word with its affixes ليناقشوهم

Antifix	**Prefix**	**Core**	**Suffix**	**Postfix**
ل	ي	ناقش	و	هم

Light stemming may affect the performance of the summarization process, especially when computing the semantic similarity between sentences and extracting nouns and verbs from a sentence.

3.1.5. Indexing

The goal of indexing in automatic summarization select tokens to describe the content of a document. Figure 3.2 shows the process of indexing a document. The process starts by selecting the document from the documents set; at first step the information of the document will indexed including title, content, size, location, and date. The process continued by splitting the indexed document into sentences using delimiters (e.g. full–stop, question–mark, exclamation–mark). Finally, those sentences will be split into tokens based on delimiters (e.g. white space). The tokens will be indexed and information about each token's location, position, frequency and weight will be recorded. Oracle database is used for indexing. The indexed tokens are used in ranking the sentences based on their weighs (e.g. tf-idf, tf-df)

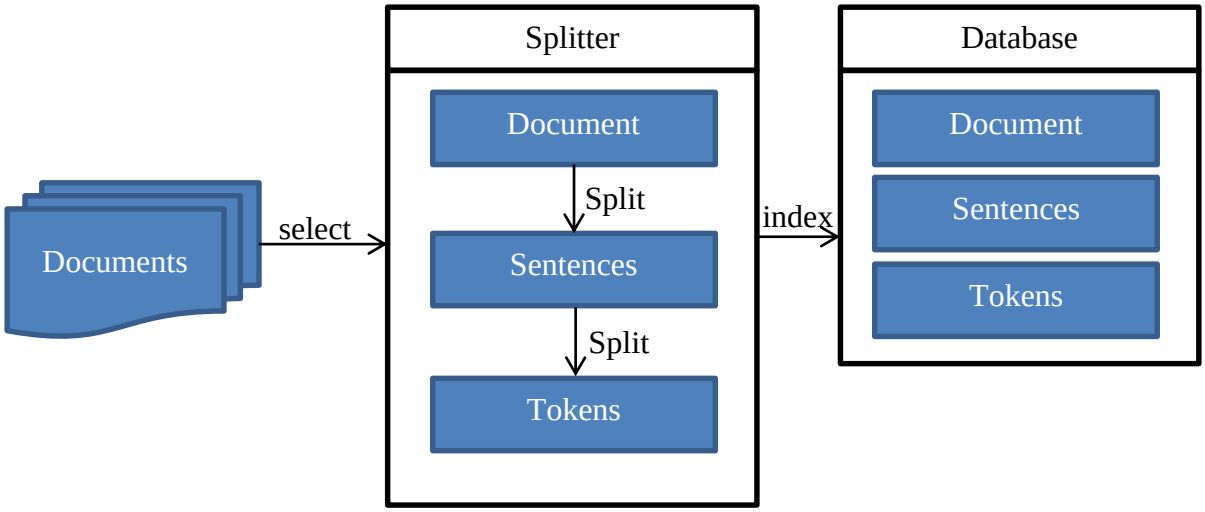

Figure 3.2: Indexing process

The next section discusses the proposed approaches and methods for multi-document summarization.

3.2. Multi-document summarization proposed methods

This section presents the methods used in multi-document summarization. The first method based on rich semantic features extracted from the sentences. The second method uses the idea of rhetorical structure theory to extract the most important sentence from each document before ranking them to generate the final summary. The third method is Cross-document structure theory (CST) for scoring sentences based on the relations with each other. Finally a hybrid of three methods.

3.2.1. Rich semantic features extraction

This approch is based on a previous research method for single document summarization. In [51], Kwaik proposed a model to automatically summarize Arabic text using text extraction. Various steps are involved in the approach: preprocessing text, extract set of features from sentences, classify sentence based on scoring method, ranking sentences and finally generate an extracted summary. The system proposed by Kwaik can be adapted for multi-document summarization

The process of the first approach contains three steps which are preprocessing of input documents, single document summarization and multi-document summarization as shown in Figure 3.3. After preprocessing step that discussed previously, the second stage perfoms a single document summarization by using multiple features of the sentences and then creating an aggregate score. This aggregated score is used to identify the most important sentences in the document. The sentences are ranked by this aggregate score for each document, after that, the most important sentences are then ranked from highest to lowest based on their aggregate scores. This is performed for each document to generate a single document summary for each of them.

The second stage is to create a multi-document summary from the ranked sentences from stage one. The creation of the summary is done by creating a new list with the top ranked sentences from each document. The final summary may contains dublicated sentences or too similar sentences, so a redundancy removal step is required.

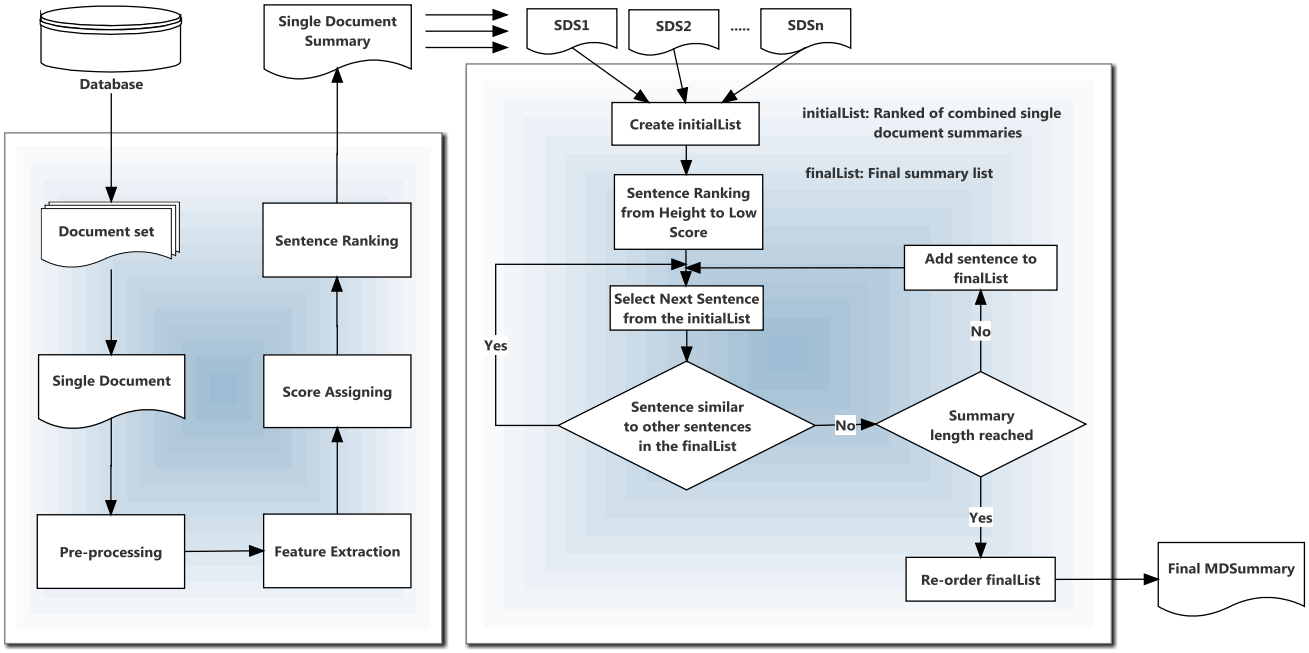

Figure 3.3: Multi-document summarization based on feature extraction

3.2.1.1. Single document summarization

The first stage concern with generating a summary for each document based on a number of features scores. In this stage each sentence is represented as a vector of features, afterword the sentences ranked based on the top aggregate feature scores sum. The features used for this stage are listed as following.

1. **Term frequency**

   Tern frequency $TF(t, d)$ is the number that the term *t* occurs in the document *d* [52]. The TF measures the importance of term t_i within the particular document d_i can be calculated by equation [53].

$$TF_{i,j} = \frac{n_{i,j}}{\sum n_{k,j}} \quad 3.1$$

   Where,

   $n_{i,j}$: The number of occurrences of the considered term (t_i) in the document d_j.

   $\sum n_{k,j}$: Sum of number of occurrences of all terms in document d_j. This summation used for normalization.

In this research, the difference variation of the terms is taken into account. For example the following terms share the same meaning, 'رئيس', 'زعيم', and 'قائد'. Another example 'شجار', 'عراك', and 'قتال'. With the help of WordNet, term synonyms could be included in the calculation of the term frequency. For each term the TF score is calculated as following:

$$TF_{i,j} = \sum_{i \in \{t_i \cup \text{synonyms}(t_i)\}} \frac{n_{i,j}}{\sum n_{k,j}} \quad 3.2$$

The number of occurrences of a term within a document measure the importance of this term in that document.

2. **Document frequency**

   This feature refers to the document frequency (DF) of a term's existence within multiple documents, or the number of documents containing the word. In general the important terms expected to appear in a few documents not in too many of them like stop words and common words.

3. **Inverse document frequency(IDF)**

   The IDF weight of a term t can be calculated from document frequency using the formula:

$$IDF_t = log\left(\frac{N}{n}\right) \quad 3.3$$

   Where,

   N: number of documents.

   n: number of documents with term *t*.

   The IDF of a term is low if it occurs in many documents and high if the term occurs in a few documents.

4. **Term Frequency-Inverse Document Frequency (TF-IDF)**

   Which refers to the term frequency times the inverse of the document frequency. It is one of the most widely used term weighting methods to determine the importance of a term throughout the document collection. TF-IDF works by determining the relative

frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document. The reason is that the terms that appears in the most of the documents will not add more importance to the sentence. These terms may not be a stop words only, but may be a very common words.

TF-IDF calculated by using the formula

$$\text{TF-IDF} = \frac{n_{i,j}}{\sum n_{k,j}} . log\left(\frac{N}{n}\right) \quad 3.4$$

After assigning TF-IDF weight for each term, the TF-IDF used to represent the sentence as a vector of weighted terms. The summation of weights in this vector will be assigned as a TF-IDF score of the sentence.

5. **Sentence position**

In news documents the first sentences tend to be the most important sentences in the document because they usually contains the main topic that the document talk about. The following formula is used to calculate the position score

$$positionScore = \frac{(\#sentences) - (sentencePosition) + 1}{(\#sentences)} \quad 3.5$$

6. **Sentence length**

This feature is useful to filter out short sentences such as subtitles, author names and date lines commonly found in news articles [54]. The short sentence tends not to be included in summaries [55], the following equation used for length score.

$$lengthScore = \frac{\#words\ in\ sentence}{\#words\ in\ longest\ sentence} \quad 3.6$$

7. **Title similarity**

Usually the title of a document describes the content of that document, so the sentence that refers to the title but not duplicates it will be more important than other sentences. The cosine similarity measure is used to measure the similarity between the given sentence and the title.

$$\cos(S_t, S_i) = \frac{S_t . S_i}{\|S_t\| \|S_i\|}$$

$$\cos(S_t, S_i) = \frac{\sum_{j=1}^{n} S_{i,j} . S_{t,j}}{\sqrt{\sum_{j=1}^{n} (S_{i,j})^2} . \sqrt{\sum_{j=1}^{n} (S_{t,j})^2}} \quad 3.7$$

Where,

S_t: Title sentence vector

S_i: Given sentence vector

Each vector represented by its term TF-DF weight, the reason of using TF-DF instead of TF-IDF will be discussed in the evaluation chapter.

8. First sentence similarity

In news articles the first sentence contains the headline for the rest of the article. Any sentence shares information with the first sentence tend to be important also. The cosine similarity measure used for similarity as described in the previous step.

9. Named entities NE

NE is used in many applications like text summarization, text classification, question answering and machine translation systems etc. [56]. Named Entities are often seen as important cues to the topic of a text. They largely define the domain of the text [57]. Named entity consists of number of categories which are: person, location, date, organization, and address. The NE score calculated by the following formula

$$NEScore = \frac{\#NE\ in\ sentence}{\#all\ NE\ in\ document} \quad 3.8$$

10. Date, time, and numbers

The sentence contains dates, time, or numbers are important to the summary because it mentions facts and events about the main topic of the documents, so that, it is important to be included in the summary. The following formulas is used to calculate these features.

$$DTNScore = \frac{\#dates, time, numbers\ in\ the\ sentence}{\#words\ in\ the\ sentence} \quad 3.9$$

3.2.1.2. Scoring and ranking

The score for each sentence is the summation of the sentence features computed in last section. This score measures the importance of the sentence.

$$Score(s) = \sum_{i=1}^{12} F_i \qquad 3.10$$

Where,

i: Feature number from 1 to 12 features.

F_i: Feature score (value)

After scoring each sentence with its features summation, the sentences ranked from the highest to the lowest aggregate score.

Section 4.4.1 reviews the impact of the use of the IDF and the DF separately along with the rest of the other features.

3.2.1.3. Summary generation

At this step we have ranked sentences based on their aggregate score values. The summary generation is done by selecting the top sentence from the ranked sentences list and add it to the output summary list. The process continues for each sentence in ranked list until reaching the summary maximum length. To avoid any redundant information, before add the sentence to the output summary list we measure its cosine similarity against the previously added sentences, if it less than a predefined threshold then add it to the output list otherwise ignore it.

3.2.1.4. Multi-document summary generation

At this step, we collect all single document summaries generated previously and put them in a new list and again rank them from the highest to the lowest aggregate score. The new list then treated like a single ranked sentences list and the summary generation is done like in the single document summarization generation step. Finally, the sentences re-ordered based on their location on the document and the published date.

3.2.2. Cross document relationships

This section describes the main contribution of this research. Our method consists of three main steps which are CST training, relation identification, and summary generation. Figure 3.4 describes the structure of the proposed method.

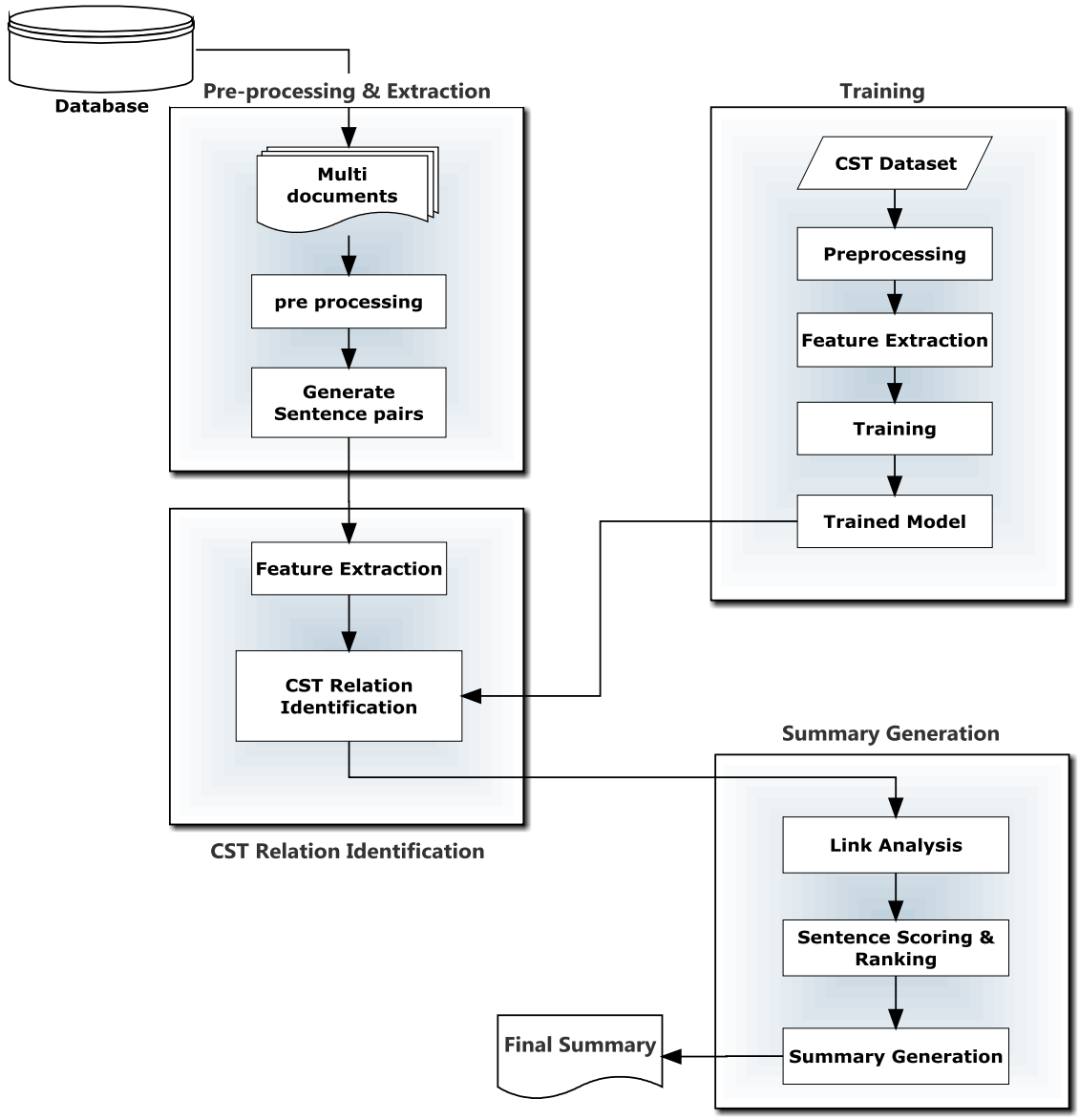

Figure 3.4: Multi-document summarization based on cross document relationships

3.2.2.1. Pre-processing and feature extraction

This step first applies a basic preprocessing on documents as mentioned in the preprocessing section. After preprocessing each document will segmented into sentences. In order to identify the CST relations in the document set, we generate a sentence pairs for all the documents, every two sentences in the sentence pairs must be related to each other, the sentence pairs generation use Algorithm 3.2. The selection of candidate sentence pair depends on their word overlap measure, the value of the word overlap is between 0 and 1, the value will be closer to 1 if the two sentences have more words in common, and when the two sentences share few words the overlap value will be closer to 0. We set a threshold value which is 0.1 the sentence pair will be added to the candidate list if their overlap measure greater than or equal to the threshold.

Algorithm 3.2: sentence pairs generation algorithm

---

input: Document set *DS*

output: Related sentence pairs

foreach Document D_i in *DS* **do**

    Split D_i into sentences;

end

foreach Document D_i in *DS* **do**

    foreach Sentence S_i in D_i **do**

        foreach Sentence S_j in other document $D <> D_i$ **do**

            $overlap = wordOverlap(S_i, S_j) = \frac{S_i \cap S_j}{S_i \cup S_j}$

            If overlap >= 0.1 **then**

                Add (S_i, S_j) to the sentence pairs list

            end

        end

    end

end

return pairs list;

The sentence pairs then processed to extract several features which will be used by the classifier to assign them a CST relationship type.

3.2.2.2. Automatic Identification of CST Relations

This research deals with four types of CST relationships which they are Identity, Subsumption, Description and Overlap, because their frequency on the document higher than the rest of relations [60]. Table 3.6 shows the description of each of the CST relations. The first column holds the name of the relationship, and the second column gives brief description. The table also shows an example for each relationship type.

Table 3.6: Description and examples of CST relations used in this research

Relationship	Description
Identity	The same text appears in more than one location.
Sentence 1	تم انتخاب توني بلير لولاية انتخابية ثانية اليوم
Sentence 2	اليوم تم انتخاب توني بلير لفترة انتخابية ثانية
Subsumption	S1 contains all information in S2, plus additional information not in S2.
Sentence 1	كانت آخر مرة سمعت فيها ايرباص A330-200 عبر الإذاعة في 22:30 بالتوقيت المحلي (01:30 بالتوقيت العالمي) يوم 1 يونيو واختفت من على شاشة الرادار حوالي 190 ميلا 306] كيلومترا [قبالة الساحل البرازيلي.
Sentence 2	اخر مرة تم السماع فيها عن ايرباص 200 - A330 عن طريق الإذاعة في الساعة 22:30 بالتوقيت المحلي (01:30 بتوقيت جرينتش).
Description	S1 describes an entity mentioned in S2.
Sentence 1	مبنى بيرلي يقع وسط مدينة ميلانو بالقرب من محطة القطارات المركزية ويضم مكاتب حكومية
Sentence 2	تصاعدت اعمدة الدخان من **مبنى بيرلي** المكون من 30 طابق ولم ترد انباء عن وقوع اصابات
Overlap	S1 provides facts X and Y while S2 provides facts X and Z.
Sentence 1	تحطمت طائرة صغيرة في الطابق ال 25 من ناطحة سحاب في وسط مدينة ميلانو وتصاعدت منه السنة اللهب والدخان
Sentence 2	تحطمت طائرة صغيرة في اطول ناطحة سحاب في مدينة ميلانو مما اسفر عن تحطم اجزاء كبيرة من الطوابق العليا

To classify the sentence pairs we need to train a classifier using annotated CST sentences pairs dataset. For Arabic language, there is no work study the use of CST for text summarization. Most of the dataset available on the web are for English [61] and Brazilian [62] languages.

We have created an Arabic annotated dataset by translating the CSTBank dataset created by Radev [61]. We used a neural network classifier to determine the CST relationships between two sentences pair. The classifier trained using the following feature extracted from the sentence pairs.

1- Cosine similarity

The cosine similarity is used to measure how the two sentences are similar

$$\cos(S_1, S_2) = \frac{\sum_{j=1}^{n} S_{1,j} . S_{2,j}}{\sqrt{\sum_{j=1}^{n} (S_{1,j})^2} \cdot \sqrt{\sum_{j=1}^{n} (S_{2,j})^2}}$$

Where

S_1, S_2: are a term frequency vector of the sentence pair.

2- Word overlap

This feature will measure the word overlap between the two sentences regardless of their frequency in the sentence. Jaccard similarity coefficient used to measure word overlap.

$$wordOverlap = \frac{\|S_1 \cap S_2\|}{\|S_1 \cup S_2\|} \qquad 3.11$$

3- Length difference

This feature measures the length difference between the sentence pair

$$LengthDiff = |length(S_1) - length(S_2)|$$

The value of length diff will indicate which sentence has more information, in case of identity the value will be closer to 0. In case of Subsumption, description, and overlap relationships, the value will be larger than 0.

4- Sentence Pairs Overlap

This feature measure the degree of relatedness of the sentence to the other sentence. It will be helpful in indicating the overlap and Subsumption relationships.

$$S_1 Overlap = \frac{\|S_1 \cap S_2\|}{\|S_1\|} \quad 3.12$$

$$S_2 Overlap = \frac{\|S_1 \cap S_2\|}{\|S_2\|} \quad 3.13$$

5- Bigram Overlap

To take the order of words into account we generate a bigram list for each sentence and repeat the overlap features for the bigram sentence pairs, the word order may be important especially for identity and Subsumption relationships.

To find the CST relationships between sentences, machine learning classifier is used. The basic idea of machine learning is to automatically learn from training data so as to be able to produce a useful output in new cases. Neural Network classifier is used as training model.

The model is trained using java neural network framework called Neuroph. We used a multi-layer feed-forward network, with the most popular back propagation learning algorithm. The number of hidden nodes is determined as following, the number of hidden node is initially set to 1. The network accuracy is then recorded for the hidden nodes after training it. Then the number of hidden nodes are incremented and the process continues. The process is repeated for a certain number of times, each time the accuracy of the network is recorded. After that, the network with the highest accuracy is chose as a training model. The network is then tested with the test data to measure its performance. In our experiments, the final parameters used for the network are as following: learning rate equals 0.2, and hidden nodes equal 8 nods. The used network model is shown in Figure 3.5, the figure indicates that 7 nodes (Layer 1) are used as input for features ($\cos(S_1, S_2)$, $wordOverlap$, $LengthDiff$, $S_1 Overlap$, $S_2 Overlap$, $S_1 BigramOverlap$, and $S_2 BigramOverlap$), 8 hidden nodes (Layer 2), and 4 output nodes (Layer 3) for CST relationship type (Identity, Subsumption, Overlap, and Description).

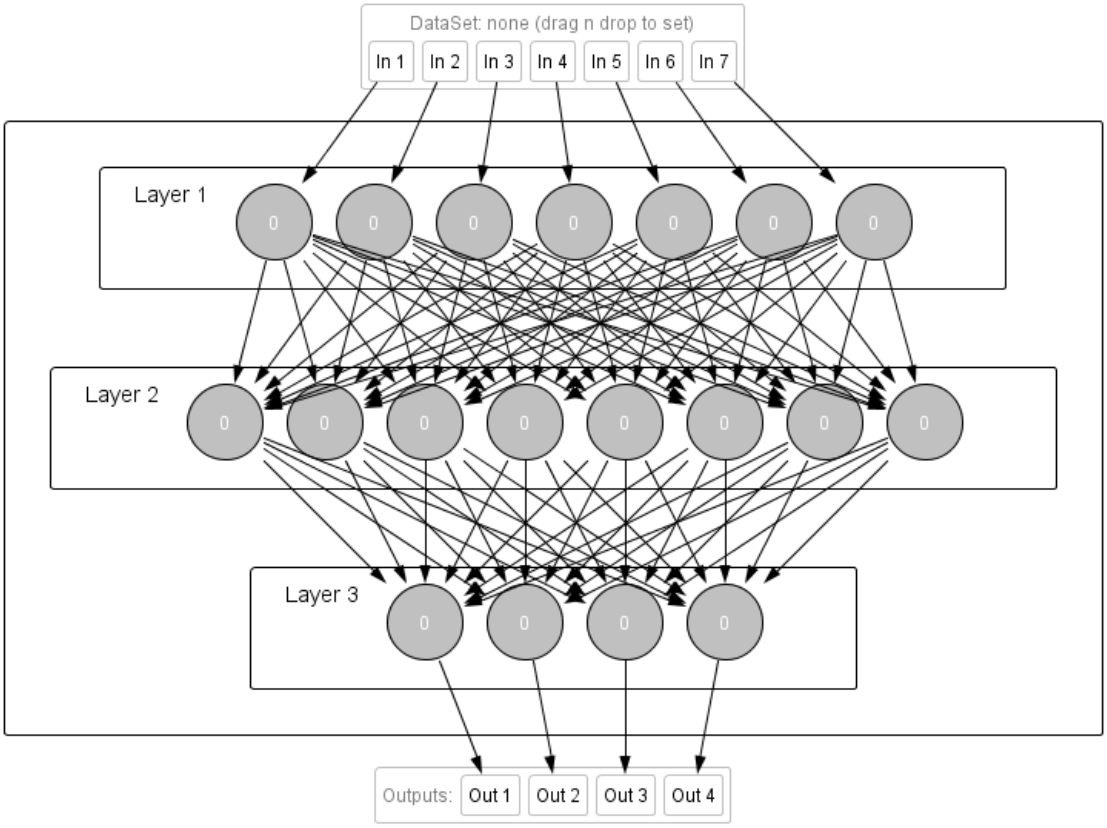

Figure 3.5: Network model for CST relationship classification

The directionality of the CST relationships depends on the type of that relation. In this research we follow some rules for assigning the directionality as follow:

Overlap: is a 2-way direction since the two sentences elaborate to each other with additional information

Description: is a 1-way direction from the second sentence which containing the description to the first sentence, because the first sentence contains the main information.

Subsumption: is a 1-way direction; from the second sentence to the first sentence since the information in the second sentence contained in the first sentence plus other additional information not in sentence one.

Identity: is a 2-way direction since the two sentences are identical.

3.2.2.3. Graph Construction and Link Analysis

After classifying sentence pairs CST relationships, the next step is to build a graph representing the relationships between sentences. The nodes in the graph represents the sentences and the links represents the relationship type. Figure 3.6 shows an example of generated graph. The direction of the relation is important in the link analysis, the nodes that has many other nodes refer to, are be important to be included in the final summary.

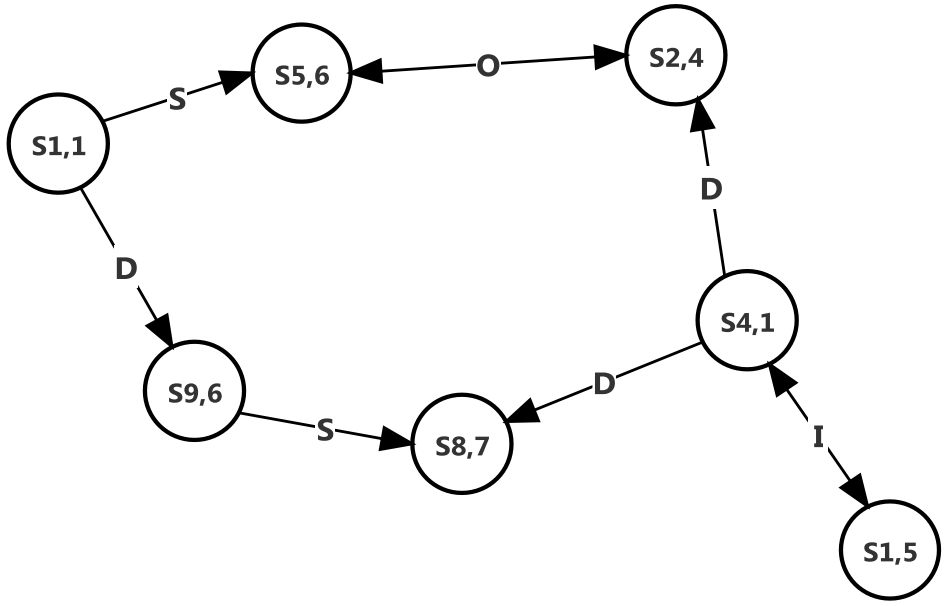

Figure 3.6: CST relationships sample graph

In addition to the directed CST relationships links, we added bi-directed links between nodes that do not have CST relationship. The new links represent the similarity between the sentences (nodes). This similarity is computed using words TF-IDF score. The final graph will be as follow $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ where $\mathbf{G}$ is a weighted directed graph. $\boldsymbol{V}$ is a set of nodes (vertices) represented by sentences. $\boldsymbol{E}$ is a set of edges (links). Given two nodes (Sentences), if there is a CST relationship between them, then $\boldsymbol{E}$ will be a directed link. If there is no CST relationship between them and their cosine similarity greater than a given threshold, then $\boldsymbol{E}$ will be bi-directed link between the two nodes. By integrating cosine similarity in the graph construction we avoided the unexpected results if the classifier fails in classifying the CST relationships between sentences because the graph will not contains enough linked sentences.

After the construction of the graph, we make some modification on the identity and Subsumption relationships links. The link modification is as following: given $\mathbf{Rel_{type}}$ a relationship type between two sentences. If $\mathbf{Rel_{type}}$ = identity then replace one sentence by the other. If $\mathbf{Rel_{type}}$ = Subsumpition, then remove the subsumed sentence. This step will improve the elimination of the redundant information. For example, the graph in Figure 3-5 is modified as following:

- Sentence pair $(S_{4,1}, S_{1,5})$ has identity relationship so remove one node of them from the graph.
- Sentence pair $(S_{1,1}, S_{5,6})$ and $(S_{9,6}, S_{8,7})$ has Subsumption relationship, $S_{1,1}$ subsumed by $S_{5,6}$ and $S_{9,6}$ subsumed by $S_{8,7}$ so we remove $S_{1,1}$ and $S_{9,6}$ from the graph.

The new graph after link modification will be as in Figure 3.7.

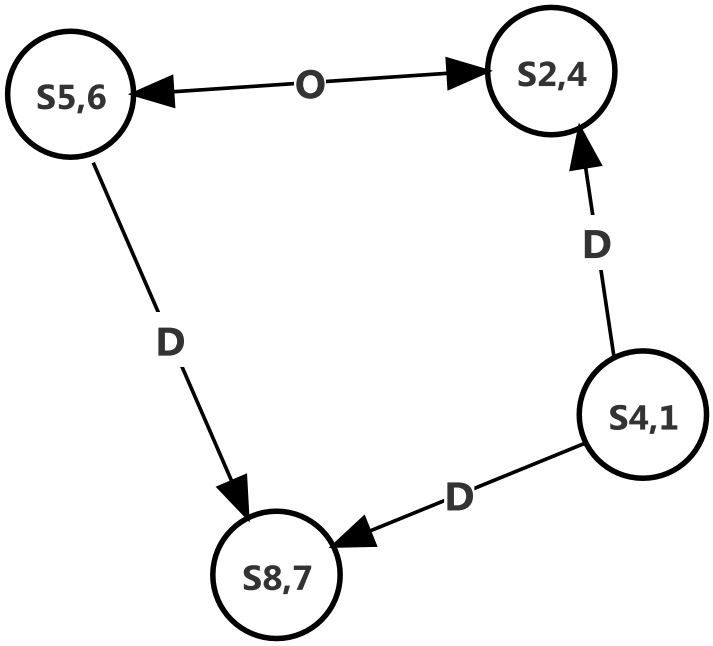

Figure 3.7: CST relationships graph after link modification

3.2.2.4. Sentence Scoring and Summary Generation

After the construction of the graph. The next step is to give a score for each node in the graph. We compute the score for each node using PageRank algorithm. PageRank is one

of the most popular link analysis algorithms and is used for web page ranking. It determines the importance of a node within a graph, based on information drawn from the graph structure. The sentences that have a large PageRank value are important to be included in the final summary. The PageRank equation is as follows:

$$PR(S_i) = \frac{1-d}{N} + d. \sum_{S_j \in M(S_i)} \frac{w_{j,i} * PR(S_j)}{L(S_j)} \qquad 3.14$$

Where $\boldsymbol{S_i}$ is the sentence under consideration. $\boldsymbol{M(S_i)}$ is the set of sentences that link to S_i. $\boldsymbol{L(S_j)}$ is the number of outbound links on sentence $\boldsymbol{S_j}$. N is the total number of sentences. $w_{j,i}$ represent the "strength" of the connection between the two vertices $\boldsymbol{S_i}$ and $\boldsymbol{S_j}$. and $\boldsymbol{d}$ is a damping factor set to 0.85. The value of $\boldsymbol{w_{j,i}}$ depends on the type of the link, if the link hold no CST relationship, then $\boldsymbol{w_{j,i}}$ is equals the cosine similarity between the two nodes otherwise it equals one. The score value depends on how many links connected to the sentence. Therefore, the more links connected to the sentence, the more important the sentence is.

Based on generated graph, we order the sentences based on their scores from the highest to the lowest score. The scored sentences then added to the summary list one by one until reaching the required summary length. While adding the sentences in the summary we check if the new sentence to be added has overlap relationship with the previously added sentences, if there is a relation then the two sentences needs to be merged to avoid redundant data. Section 3.2.2.5 describes our novel approach for redundancy removal.

After the generation of the summary, the system reorder the sentences to make the summary more readable. The reordering is done based on four features which are: similarity with title, sentence position, publish date, and related sentences positions. The first two feature computed as in section 3.2.1.1. The publish date is available for each document. The last feature is computed using the constructed graph, for each sentence, the system extracts all related sentences to the sentence under consideration from the graph, then the feature value is computed by taking the sentence position feature mean value for all of them. If the sentence is important then it is expected that all related

sentences have a position score closer to one. Finally, the sentences are reordered based on publish date and features aggregate score from the highest to the lowest score.

3.2.2.5. Redundancy Removal

As mentioned in section 1.3; redundancy removal one of the main challenges in multi-document summarization. Redundancy removal aims to reduce the recurrence of information. Referring back to section 3.2.2.3; we note that the links modification we have done before, reduced a lot of possible redundant information by replacing nodes with each other. But there are still nodes hold redundant data, these nodes are the nodes with overlap relationship between them. This section discusses our method for removing redundant data between overlapped sentences.

Given two sentences $\boldsymbol{S_1}$ and $\boldsymbol{S_2}$, the algorithm will merge them and produce a new sentence without redundant information. To illustrate the proposed method, for example: suppose the following two sentences to be included in a summary.

S_1: ذهب محمد برفقة والدته إلى السوق، ليشتريا الفاكهة

S_2: ذهب محمد الى السوق، وقابل صديقه حسام هناك

It is clear that there are similar span of text between them, so including both sentences in the summary will produce a redundant text. One solution is to choose just one of them based on some feature score like in section 3.2.1.1. Choosing one sentence will lead to ignore important information in the other sentence. For the given example the optimal solution is to merge the two sentences as following:

'ذهب محمد برفقة والدته إلى السوق، ليشتريا الفاكهة، وقابل صديقه حسام هناك'

The new sentence is a combination of the two sentences. The proposed algorithm for redundancy elimination consists of three steps:

1- Sentence Splitting.

At this step, each sentence is splitted into smaller units. These units are noun phrases NP and verb phrases VP. For example the above sentences will be represented as following:

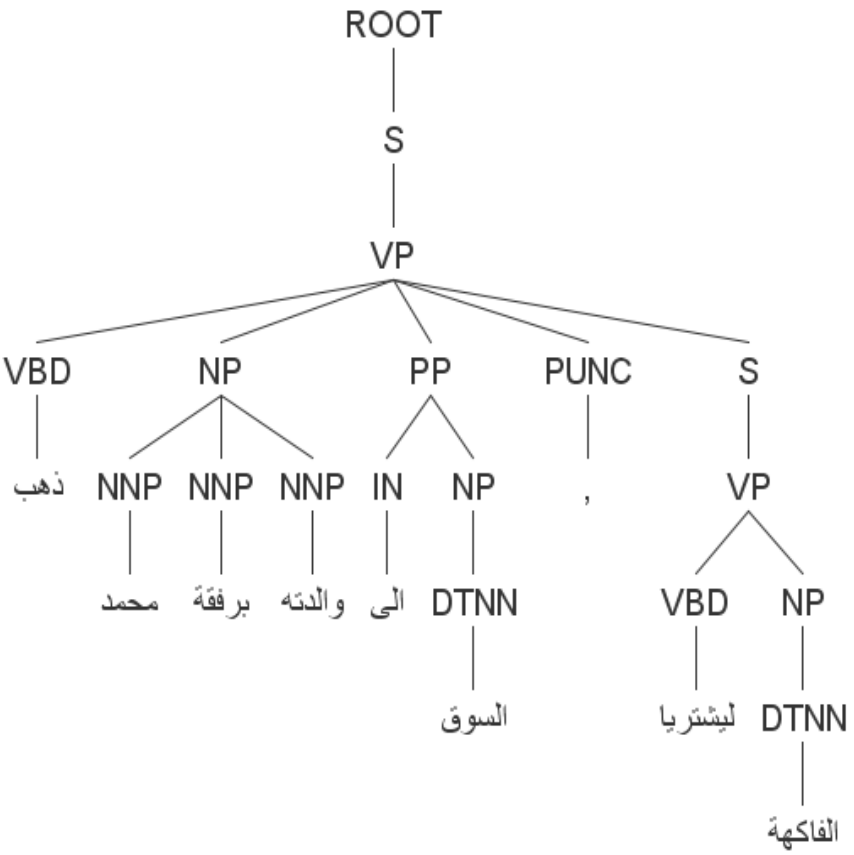

Figure 3.8: Parse tree for S1

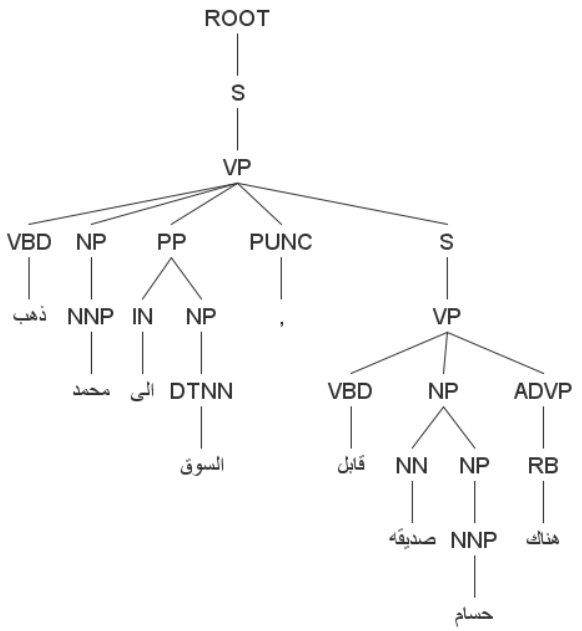

Figure 3.9: Parse tree for S2

Figure 3.8 and Figure 3.9 represents a parse tree for sentences S1 and S2 respectively. Sentence S_1 consists of two verb phrases which are 'ذهب محمد برفقة والدته إلى السوق', and 'ليشتريا الفاكهة'. S_2 also consists of two verb phrases which are 'ذهب محمد إلى السوق', and 'قابل صديقة حسام هناك'. We used Stanford statistical parser [64] to parse the sentences. Section 4.2.2 describes sentence parsing in details.

2- Units Similarity.

After splitting each sentence and building a parse tree for each one. The next step is to identify similar units between them. Similar units are those that share many words between them. Cosine similarity measure is one of the most common used measures to score the similarity between two texts. Algorithm 3.3 describes the process of selecting the similar units between two sentences.

Algorithm 3.3

input: $S_{1,units}, S_{2,units}$

output: Similar units pairs

foreach Unit *x* in $S_{1,units}$ **do**

  for All units in $S_{2,units}$ **do**

    find a non-chosen unit *y* that have maximum cosine similarity with *x*, such that CosSim(*x*, *y*)>=50%

    if unit *y* founded **then**

      mark *y* as chosen unit

    end

  end

end

return pairs list;

Table 3.7 shows the cosine similarity for each units pairs. From the table it is clear that the units in first row will marked as similar units.

Table 3.7: Cosine Similarity for units pairs

Units pairs	Cosine Similarity
('ذهب محمد الى السوق', 'ذهب محمد برفقة والدته إلى السوق')	**51.6%**
('وقابل صديقه حسام هناك', 'ذهب محمد برفقة والدته إلى السوق')	0%
('ذهب محمد الى السوق', 'ليشتريا الفاكهة')	0%
('وقابل صديقه حسام هناك', 'ليشتريا الفاكهة')	0%

3- Final Candidate Units.

This step will loop through the similar units pairs list and for each pair it will replace one unit by the other. The replacement process depends on the overlap ratio of words from the first unit in the second unit, and vice versa. The unit that has the larger overlap ratio will replaced by the other unit and removed from the list. For example, taking the first units pair in Table 3.7, the first unit overlap ratio equal 0.67 while the second unit overlap ratio equal 1. it is clear that the second unit e.g. ' ذهب محمد إلى السوق' has the larger overlap ratio which is 100% because all of its tokens are in the

first unit. As a result the second unit will be removed from its sentence. The new sentences will be as following:

S_1: ذهب محمد برفقة والدته إلى السوق، ليشتريا الفاكهة

S_2: وقابل صديقه حسام هناك

4- Re-align units and form the result sentence.

Finally the chosen units from each sentence are re-ordered to form the final sentence. We follow simple method to re-order the units. To make it more readable, the new sentence is formed depending on the sentence that contains the largest number of units. From the previous example S_1 has two units while S_2 has one unit, so the final sentence is formed by ordering units from S_1 followed by units from S_2. The final sentence is as follows:

'ذهب محمد برفقة والدته إلى السوق، ليشتريا الفاكهة، وقابل صديقه حسام هناك'

Chapter 4

4. Evaluation and Results

This chapter describes the experimental results of the proposed approaches. Section 4.1 describes the programming language and tools used to develop the proposed approaches. Section 4.2 describes the datasets used for training and evaluation. The evaluation method is discussed in section 4.3. Finally section 4.4 discuss the evaluation results for the proposed methods and compare them with other methods.

4.1. Tools

For implementation, Java programming language is used to implement the proposed methods, because it contains many libraries for text preprocessing.

4.1.1. AraNLP

AraNLP is a free, Java-based library that covers various Arabic text preprocessing tools. Although a good number of tools for processing Arabic text already exist, integration and compatibility problems continually occur. AraNLP is an attempt to gather most of the vital Arabic text preprocessing tools into one library that can be accessed easily by integrating or accurately adapting existing tools and by developing new ones when required. The library includes a sentence detector, tokenizer, light stemmer, root stemmer, part-of speech tagger (POS-tagger), word segmenter, normalizer, and a punctuation and diacritic remover [63].

4.1.2. The Stanford Parser

A natural language parser is a program that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb. Probabilistic parsers use knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of new sentences. These statistical parsers still make some mistakes, but commonly work rather well. Their development was one of the biggest breakthroughs in natural language

processing in the 1990s. we used the Stanford parser in order to tokenize the sentences into its small components e.g. noun phrases and verb phrases at the redundancy elimination step. Here is an example of parsed sentence using Stanford parser [64].

Sentence: الرجل السعيد يسعد الناس

Parse String:

```
(ROOT
  (S
    (NP (DTNN الرجل) (NNP السعيد))
    (VP (VBP يساعد)
      (NP (DTNN الناس)))))
```

Parse tree

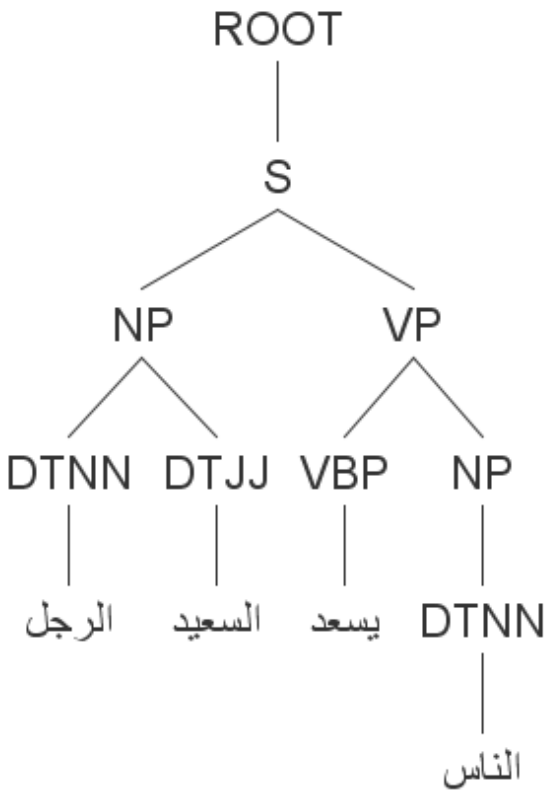

Figure 4.1: Parse tree for sentence "الرجل السعيد، يسعد الناس"

Sentence: أعلن المجلس الأمني القومي الإيراني انه لن يفرج عن مشاة البحرية البريطانيين

Parse String:

```
(ROOT
  (S
    (VP (VBD اعلن)
      (NP (DTNN المجلس) (DTJJ الامني) (DTJJ القومي) (DTJJ الايراني))
      (NP
        (NP (NN انه))
        (SBAR
          (S(VP
              (PRT (RP لن))
              (VBN يفرج)
              (PP (IN عن)
                (NP (NN مشاة)
                  (NP (DTNN البحرية) (DTJJ البريطانيين)))))))))))
```

Parse Tree:

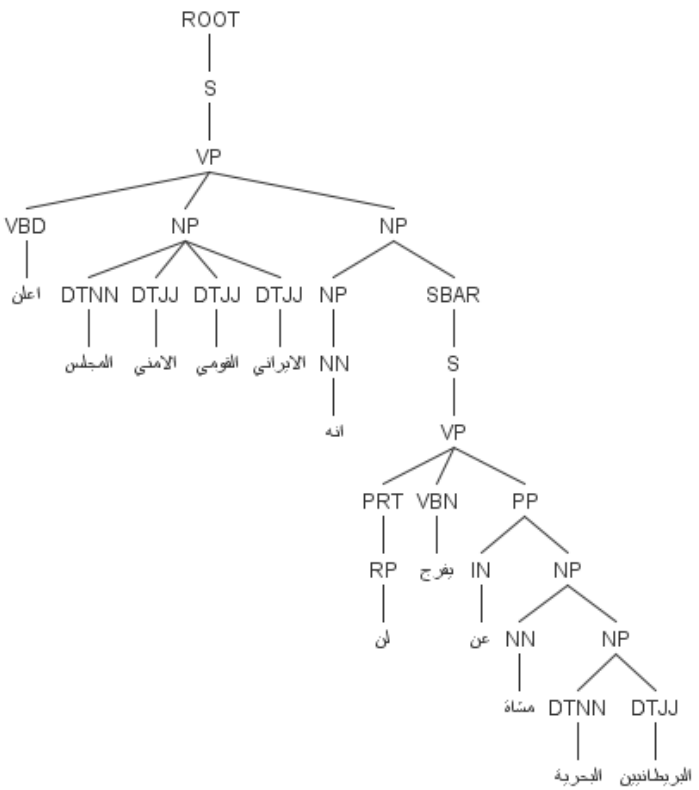

Figure 4.2: Parse tree for sentence " أعلن المجلس الأمني القومي الإيراني انه لن يفرج عن مشاة البحرية البريطانيين"

Figure 4.1 and Figure 4.2 represents the parse trees for two different sentences. In each figure, the "**Sentence**" refers to the sentence to be parsed, "**Parsing String**" is the output of the parser when parsing the string, and "**Parse Tree**" is a visualization of the "Parsing String". Each tree shows the syntactic structure of the sentence. Each node in the tree is either a root node, a branch node, or a leaf-terminal node. A root node is a node that doesn't have any branches on top of it. Within a sentence, there is only ever one root node. A branch node is a mother node that connects to two or more daughter nodes. A leaf node, however; is a terminal node that does not dominate other nodes in the tree.

Taking Figure 4.1 as example, S is the root node, NP and VP are branch nodes for noun and verb phrases, and الرجل (DTNN), السعيد (DTJJ), يسعد (VBP), and الناس (DTNN) are all leaf nodes. The leaves are the lexical tokens of the sentence. A node can also be referred to as parent node or a child node. A parent node is one that has at least one other node linked by a branch under it. In the example, S is a parent of both NP and VP. A child node is one that has at least one node directly above it to which it is linked by a branch of a tree. From the example, الرجل is a child node of DTNN parent node. The branch nodes are represented by either part of speech tags or chunk tags.

POS tags are assigned to a single word according to its role in the sentence. Chunk tags are assigned to groups of words that belong together (i.e. phrases). The following abbreviations are used in the previous examples

VP, NP, **PP**, **NN**, **VBN**, **DT**, **JJ, IN, DTNN**: DT+ NN, and **DTJJ**: DT + JJ

4.2. Datasets

4.2.1. CSTBank dataset

CSTBank is a corpus of document clusters manually annotated for CST relationships. It contains clusters of documents created in a variety of ways (e.g. manually and automatically clustered documents) and is organized by families, which describe the text sources and clustering methods used to group documents by their respective topics. The CSTBank dataset available only in English language. Since this thesis aims to summarize multi-documents texts in Arabic. We created a new CST dataset in Arabic language. For

this purpose, we have translated the dataset mentioned above from English to Arabic with help of a human translator to get optimal translation. The dataset consists of nine documents talking about the crash of a small plane into a skyscraper in Milan, Italy that occurred on April 18, 2002 and the events surrounding it. They were collected live from the Web and were published by five different news sources. Each document is presented with its source and time of publication. The CST relationships between sentences were annotated by two judges working independently. The first and the second judges were annotated about 774, and 672, respectively, CST relationships between sentences. As in this research we just used four CST relationships which are: "Identity", "Description", "Overlap", and "Subsumption", we pick a part of the dataset to use it for training and testing.

4.2.2. TAC 2011 MultiLing Pilot dataset

For multi-document summarization we used the TAC 2011 MultiLing Pilot dataset [65]. It is a multilingual dataset of seven languages (Arabic, Czech, English, French, Greek, Hebrew, Hindi). The original text document were extracted from WikiNews website, which covers a variety of news topics. The creation of the corpus was started with translating the English documents to each of the other languages: Arabic, Czech, French, Greek, Hebrew, Hindi. The dataset contains 100 document classified into 10 clusters (i.e. document set). Each document set provide information about an individual news event.

For each document set, three model summaries are provided by fluent speakers of each corresponding language (native speakers in the general case). A number of 12 people participated in translating the English corpus into Arabic and in summarizing the set of related Arabic articles. Each document set summary size is between 240 and 250 words.

Beside the document sets the creator of the dataset provided a number of peer summaries for evaluation. For the Arabic language, there were 7 participants (peer summaries) ID1, ID2, ID3, ID4, ID6, ID7, and ID8. In addition to two baseline systems, one acting as a global baseline (System ID9) and the other as a global topline (System ID10). The baseline systems show better evaluation result more than the others peer systems. These two systems are described briefly in the following section

Baseline/Topline Systems

Global baseline system (ID9), represents each document in the document set as vector space using a bag of word approach, then find the centroid of that space. The system then select the texts that is most similar to the centroid (based on cosine similarity). If the text exceeds the summary word limit, then only a part of it is used to provide the summary. Otherwise, the whole text is added as summary text. If the summary is below the lower word limit, the process is repeated iteratively adding the next most similar document to the centroid.

The global topline system (ID10) uses the model summaries as a given. Then the documents are represented using n-gram graphs and merged into a representative graph [66]. Then, an algorithm produces random summaries by combining sentences from the original texts. The summaries are evaluated by their MeMoG score with respect to the model summaries.

4.3. Evaluation metrics

In this research we use a will-known automatic evaluation method: recall-oriented understudy for gisting evaluation (ROUGE) [67]. Rouge automatically determines the quality of a computer generated (peer) summary through comparing it to other ideal (model) summaries created by humans. ROUGE includes five measurement metrics Rouge-N, Rouge-L, Rouge-W, Rouge-S and Rouge-SU.

ROUGE-N: N-gram based co-occurrence statistics. It measure how much two summaries are similar by counting the number of n-gram matches between these two summaries. ROUGE-N is computed as follows:

$$ROUGE - N = \frac{\sum_{S \in \{Reference\ summaries\}} \sum_{N-gram \in S} Count_{match}(N - garm)}{\sum_{S \in \{Reference\ summaries\}} \sum_{N-gram \in S} Count(N - garm)} \qquad 4.1$$

Where

$\boldsymbol{N}$ = the length of $N - gram$.

$\{\boldsymbol{Reference\ summaries}\}$ = the human summaries.

$\mathbf{Count_{match}(N-garm)}$ = the maximum number of n-grams co-occurring in a candidate –peer- summary and $\{Reference\ summaries\}$.

For example consider the following sentence 'الشرطة قتلت الرجل المسلح'. If we use ROUGE-2 measure then the previous sentence consists of three bigrams as follows {'الشرطة قتلت', 'قتلت الرجل', 'الرجل المسلح'}

ROUGE-S: Skip-Bigram Co-Occurrence Statistics

Skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps. Skip-bigram co-occurrence statistics measure the overlap of skip bigrams between a candidate translation and a set of reference translations. Using the previous example, the sentence has 6 skip-bigrams as the following {'الشرطة قتلت', 'الشرطة الرجل', 'الشرطة المسلح', 'قتلت الرجل', 'قتلت المسلح', 'الرجل المسلح'}.

ROUGE-SU: Extension of ROUGE-S

One potential problem for ROUGE-S is that it does not give any credit to a candidate sentence if the sentence does not have any word pair co-occurring with its references, for example consider the following sentences

S1: 'المسلح الرجل قتلت الشرطة'

S2: 'الشرطة قتلت الرجل المسلح'

S2 is the exact reverse of S1 and there is no skip bigram match between them. However, we would like to differentiate sentences similar to S2 from sentences that do not have single word co-occurrence with S1. To achieve this, ROUGE-S was extended with the addition of unigram as counting unit i.e. ROUGE-SU = Skip-bigram + unigram-based co-occurrence statistics. The extended version is called ROUGE-SU.

A detailed description for the others measures can be found in [67]. In our evaluation will use two metrics which are ROUGE-1 and ROUGE-2.

ROUGE-1 and ROUGE-2 compares the unigram and bigram overlap between the system summary and the human abstracts. To evaluate our system generated summaries we used three measures precision, recall, and F-measure. Precision (P) is a measure of how much of the information that the system returned is correct. It equals the "number of system correct n-grams" divided by the "total number of system n-grams". Recall (R) measures the coverage of the system and it equals the "number of system correct n-grams" divided by the "total number of human n-grams". F-measure is a measure that combines precision and recall is the harmonic mean of precision and recall. These measures are computed using the following equations:

$$R = \frac{\left\|G_{ref} \cap G_{peer}\right\|}{\left\|G_{ref}\right\|} \quad 4.2$$

$$P = \frac{\left\|G_{ref} \cap G_{peer}\right\|}{\left\|G_{peer}\right\|} \quad 4.3$$

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad 4.4$$

Where,

$\boldsymbol{G_{ref}}$ = the gram of the reference- model- summary.

$\boldsymbol{G_{peer}}$ = the grams of the peer- system- summary.

β = balance parameter. When β equal one, precision and recall are given the same weight. When β less than one, recall will have greater weight over precision. When β greater than one, precision will be favored over recall. In our experiment we will assume that β equal one, so P and R have the same weight.

4.4. Evaluation and Results

In this section we will show our experimental results for each method along with its variations, then compare and discuss the results. To evaluate our system, we used the TAC 2011 dataset which provide us with three human summaries for each document set, the dataset also contains a set of peer summaries for evaluation. The proposed system is evaluated using ROUGE and cosine similarity with human summaries. The evaluation process run as shown in Figure 4.3: given a document set and a summarization method;

the summarization method is applied on the document set producing a single summary for that set. The next step is to calculate the ROUGE score and cosine similarity between the generated summary and the model summaries. Finally we take the average values of precision, recall, F-measure, and cosine similarity. TAC 2011 provided three model summaries for each document set.

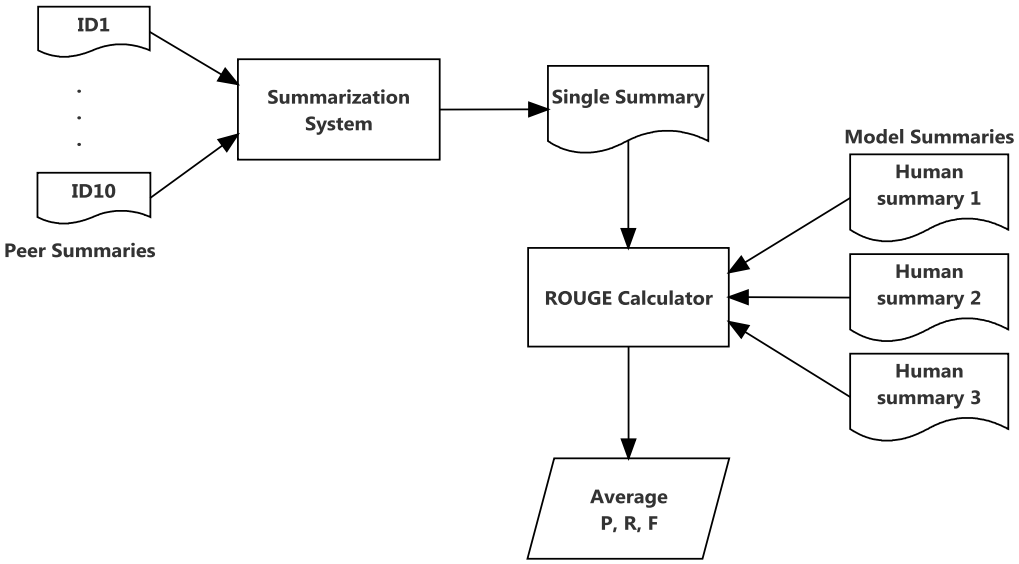

Figure 4.3: Evaluation Process

4.4.1. Features Extraction

The first summarization method in our thesis based on generating features vector for each sentence in the document set. These features scores are aggregated together to get the aggregate score for each sentence. Then the sentences that have the highest score are added to the summary until reaching the summary limit. Each sentence must not have a cosine similarity greater than 0.70 with the other sentences added previously to the summary. We assumed that each feature in the feature vector have a weight equal to 1. We have created three versions of the first method which are: FB_MDS_tfidf, FB_MDS_tfdf, and FB_MDS_RST, where **FB_MDS_** refer to "Feature based multi-document summarization". FB_MDS_tfidf method uses TF-IDF weight in addition to the

rest of features mentioned in section 3.2.1.1, while FB_MDS_tfdf method, uses TF-DF weight instead.

The next section explains each version and shows the evaluation results for each one compared to the baseline and the topline summaries.

4.4.1.1. FB_MDS_tfidf method:

In this version the term weighting is calculated by its TF-IDF. Figure 4.4 shows the ROUGE-2 evaluation results compared to the topline summary (id10) and baseline summary (id9). We note that TF-IDF has a good result compared to baseline summary. The generated summary length is equal to 250 words only. The P, R, and F refers to precision, recall, and F-measure, respectively.

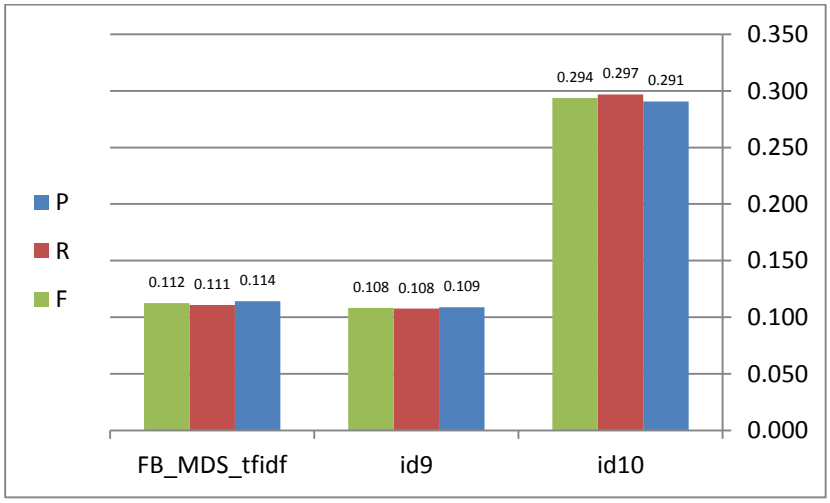

Figure 4.4: TF-IDF ROUGE-2 250 words evaluation results

Figure 4.5 shows the evaluation result of 10 sentences length summary. We note that the result is improved. This is because the 250 word summary may discard some important information from the summary. In some cases using the sentences number as a stop criteria may affect the final summary. For example, if the sentences length in the document is very small, the summary will be small compared to the human summaries. If the sentences length to too large, then the final summary will be larger than the human summaries. Using the number of words as a stop criteria will fix the length of summary compared to the human summary. In the rest of experimental results we will use the

number of words as a stop criteria, the human summaries provided in the dataset is a 250 words summaries so a 250 words summary is used in this research.

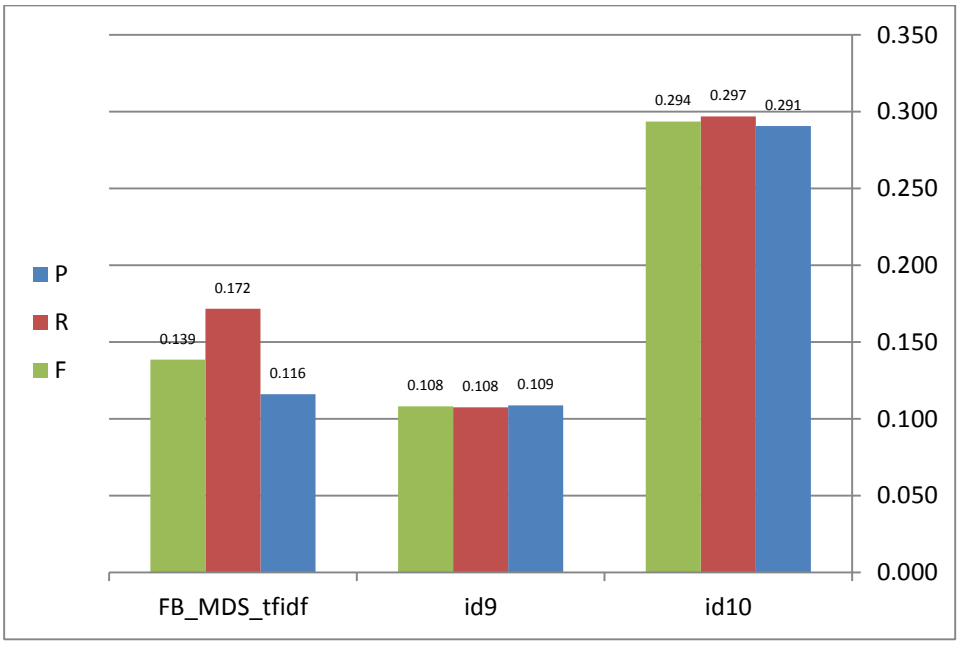

Figure 4.5: TFIDF ROUGE-2 10 sentences evaluation results

4.4.1.2. FB_MDS_tfdf:

Using TF-IDF is the standard term weighting to determine the importance of term in the dataset. A very common terms like stop words in the dataset are not important. The TF-IDF will give a common terms low or zero weight rather than rare terms which will have a height weight. But is TF-IDF doing well in multi-document summarization, especially when we deal with small document set – between 10 to 15 documents in each document set. Figure 4.6 and Figure 4.7 show ROUGE-2 evaluation results of a 250 word summary and 10 sentences using a TD-DF for term weighting. TF-DF shows better results over TF-IDF and the baseline summary (ID9), in both summarization level 250 words and 10 sentences

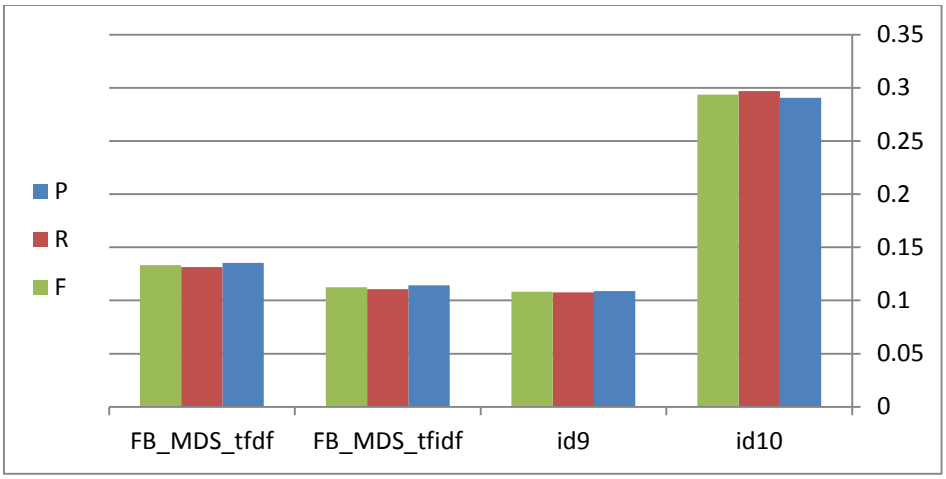

Figure 4.6: 250 words ROUGE-2 evaluation results using TF-DF

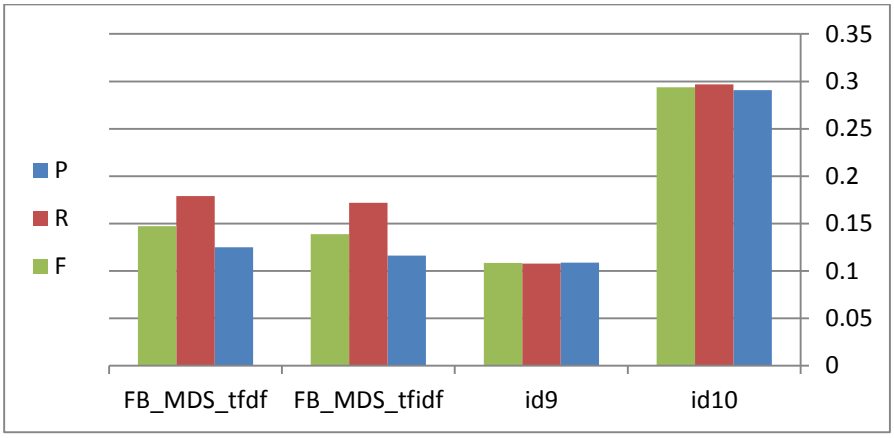

Figure 4.7: 10 sentences TF-DF ROUGE-2 evaluation results

The reason that TF-DF works better than TF-IDF is because, TF-DF gives the terms more importance. For example, suppose the following sentence from the dataset:

‘البحرية الملكية تصر على انهم كانوا يعملون في المياه العراقية’

And the sentence after removing stop words is:

‘البحرية الملكية تصر يعملون المياه العراقية’.

The previous sentence is exist in one of the human summaries, so it is important for any summarization method to give it a height score.

Table 4.1: TF-IDF, TF-DF scores for the sentence ‘ البحرية الملكية تصر على انهم كانوا يعملون في المياه العراقية’

Token	TF	DF	TF-IDF	TF-DF
البحرية	9	10	0	90
الملكية	4	5	1.204	20
تصر	1	3	0.52	3
يعملون	4	8	0.38	32
المياه	10	10	0	100
العراقية	10	10	0	100

The final normalized score for the sentence is 2.14 and 19.16 for TF-IDF, and TF-DF respectively. The reason that the TF-IDF has a low score is because that three terms have zero IDF score. Given that the number of the documents is ten, then the IDF value is zero for the ‘البحرية’, ‘المياه’, and ‘العراقية’ terms. These terms treated as a common terms like stop words.

In the context of multi-document summarization, treating those terms as common words will decrease the summarization performance, because we already remove all of the common tokens from the data set at the preprocessing step- see section 3.1. Using TF-DF instead, will add more importance to the non-common terms that appears many times in the documents.

4.4.1.3. Rhetorical structure theory (FB_MDS_RST)

The last version is a hybrid system for multi-document text summarization using features vector and rhetorical structure theory (RST). The RST is used in order to extract the most important sentences in the document,

In this method, before computing sentences features, the documents filtered to extract the main sentences from it. The document firstly represented as a binary tree based on the rhetorical relation between the sentences. At the rhetorical relationship identification process, the sentences classified into two classes which are nucleus and satellites based on a pre-defined cues. After identifying the type of each sentence only the nucleus

sentences taken for the next steps which are feature extraction, ranking, and summary generation. Paragraphs can be classified as nucleus or satellites. Nucleuses sentences are considered the most important parts of a text, whereas satellites contributing to the nucleus are secondary [58, 59].

Figure 4.8 shows the ROUGE-2 evaluation results for a 250 word summary length. The RST improves the results because the summarization process done only on the main sentences of the documents.

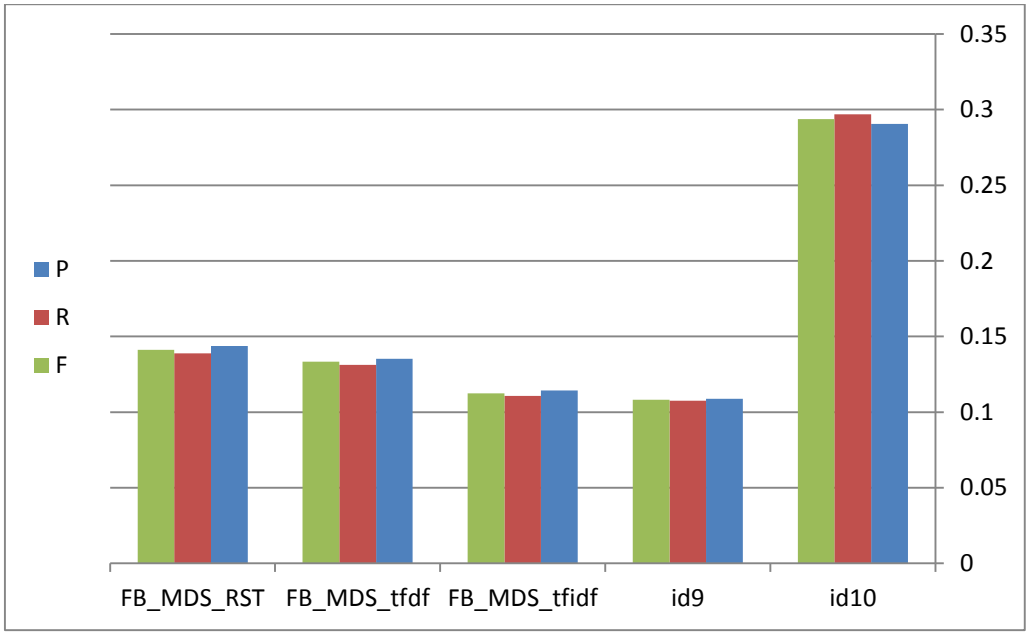

Figure 4.8: 250 word RST ROUGE-2 evaluation results

4.4.2. Method 2: CST based method

CST is used to determine the highly semantically relevant sentences in the document set. Figure 4.9 shows the ROUGE-2 evaluation results of the CST method. As shown in the figure, CST make a significant result improvement compared to RST, TF-DF, and TF-IDF methods.

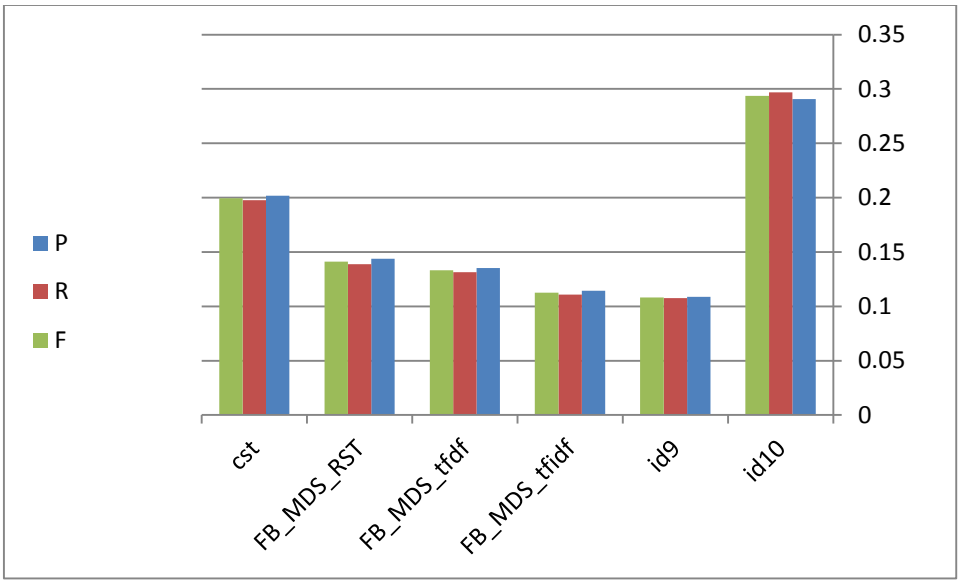

Figure 4.9: CST ROUGE-2 250 word evaluation results

Note that, in our experiments we assume that, the weight between any two nodes which have CST relationship is equal one. The CST classification process has important effect on the resulting summary. To demonstrate this, Figure 4.10 shows the precision and recall values for the CST based summaries of a nine topics from the real dataset. $\boldsymbol{T1}$ ... $\boldsymbol{T10}$ are refers to the topic id from the dataset. P and R, are refers to precision and recall respectively. It is notable that there are three topics that have poor P and R results, while other topics have good results. If the classifier failed on classifying the CST relationships between the sentences, then the performance will be decreased. As shown in Figure 4.10, topics 3, 9, and 10 have the best precision, recall, and F-measure results, which mean that, the classifier make a good CST identification for the nodes related to these topics.

The CST training data set might be the reason of the poor classification results for some documents, the reason of that might be because of the differences of the topics discussed in the training data set and the real data set. The training dataset that used in this research talks about one topic related to a plane crash. Adding more data about different topics to the training data set will improve the CST classification.

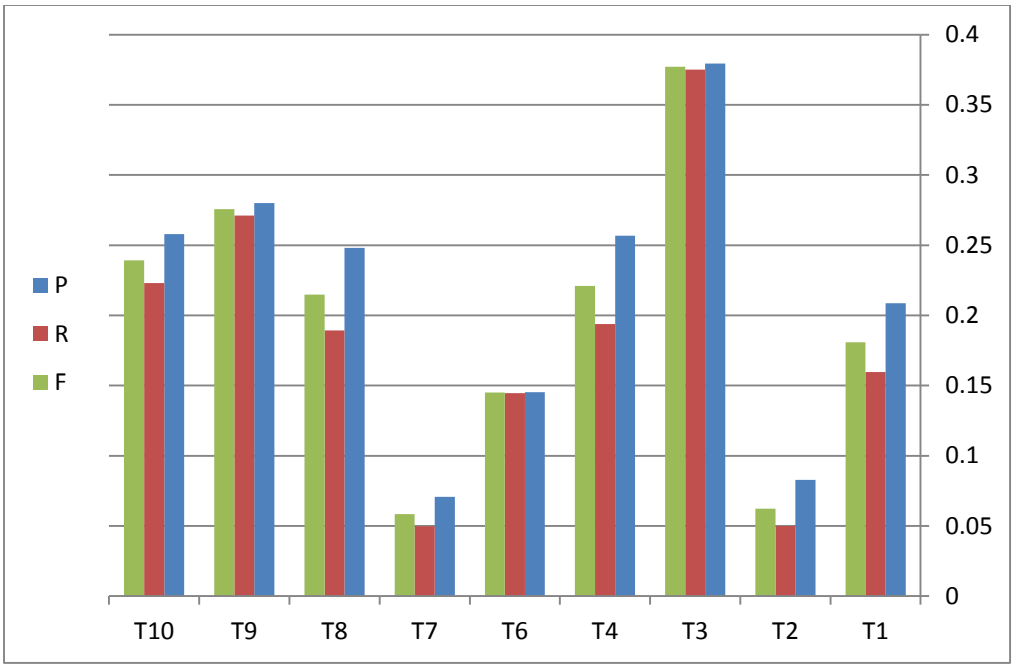

Figure 4.10: CST precision and recall evaluation result for the topics from the dataset

4.4.3. Overall system evaluation results

This section talk about the overall evaluation results for our methods and the peer summaries provided by the dataset. Table 4.2 shows the cosine similarity between proposed methods and the human summaries (H1, H2, and H3). We removed stop words and apply a light stemming before compute the cosine similarity. CST method has a better cosine similarity than the other proposed methods. Compared with peers summaries; CST is among two best peer systems (ID10 and ID4) scores.

Table 4.3 shows the effect when we apply a root stemming on the text, TF-DF is among best four peer systems (ID10, ID2, ID3, and ID4) scores and make good results compared to CST, TF-IDF, and RST methods.

Figure 4.11 and Figure 4.12 shows a graphical representation of average cosine similarity ordered from the heights to the lowest cosine cores.

Table 4.2: 250 words cosine similarity with light stemming

Method	H1	H2	H3	AVG
ID1	0.444	0.352	0.412	0.403
ID2	0.625	0.448	0.527	0.534
ID3	0.632	0.396	0.587	0.538
ID4	0.644	0.452	0.553	0.550
ID6	0.524	0.400	0.477	0.467
ID7	0.377	0.329	0.369	0.358
ID8	0.521	0.419	0.469	0.470
ID9	0.454	0.336	0.404	0.398
ID10	0.647	0.441	0.654	0.581
FB_MDS_ TFDF	0.581	0.424	0.508	0.504
FB_MDS_ TFIDF	0.499	0.368	0.457	0.441
CST	**0.636**	**0.443**	**0.557**	**0.546**
FB_MDS_ RST	0.510	0.377	0.475	0.454

Table 4.3: 250 words cosine similarity with root stemming

Method	H1	H2	H3	AVG
ID1	0.628	0.560	0.616	0.601
ID2	0.762	0.651	0.700	0.704
ID3	0.757	0.586	0.724	0.689
ID4	0.752	0.610	0.699	0.687
ID6	0.676	0.574	0.651	0.633
ID	0.6 2	0.561	0.600	0.588
ID8	0.660	0.602	0.635	0.632
ID9	0.623	0.551	0.606	0.593
ID10	0.777	0.619	0.773	0.723
FB_MDS_ TFDF	0.733	**0.609**	**0.690**	**0.677**
FB_MDS_ TFIDF	0.679	0.592	0.662	0.644
CST	**0.737**	0.602	0.687	0.676
FB_MDS_ RST	0.693	0.586	0.673	0.651

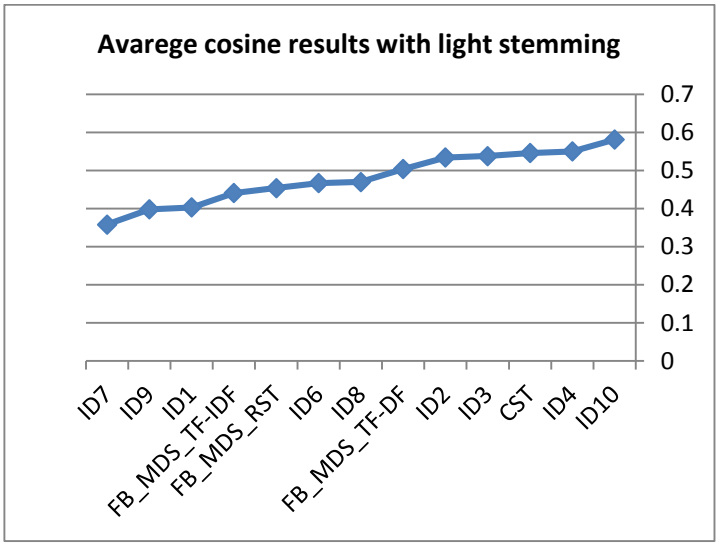

Figure 4.11: Similarity measure with light stemming

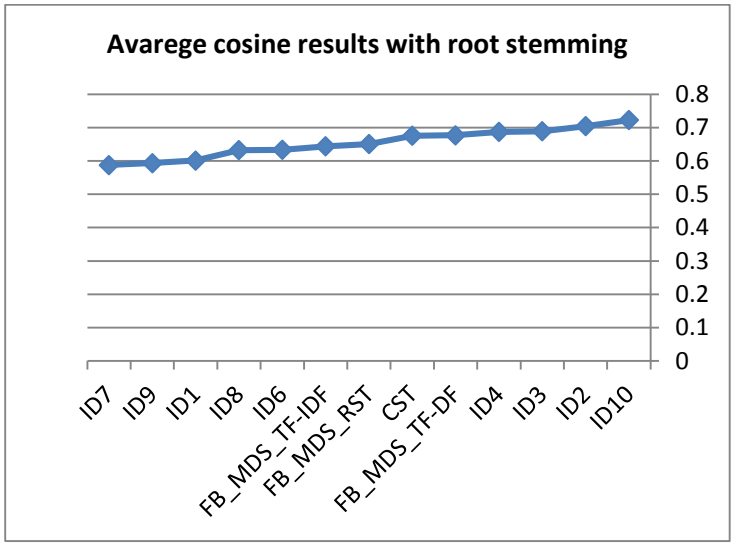

Figure 4.12: Similarity measure with root stemming

Figure 4.13 and Figure 4.14 represents the overall ROUGE-1 and ROUGE-2 results respectively. The topline summary –ID10- shows the best ROUGE-1 and ROUGE-2 results since it uses the (human) model summaries as a given to generate the final

summary. Since it uses the human summaries, the topline summary could be excluded from the comparison as it expected to give best results.

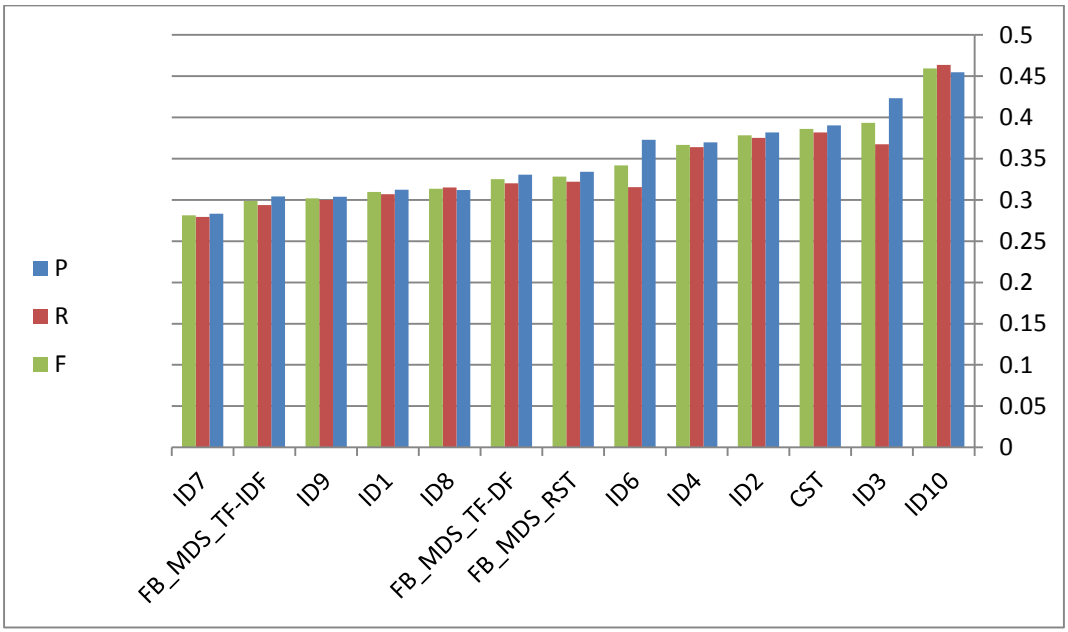

Figure 4.13: Arabic multi-document summarization ROUGE-1 results

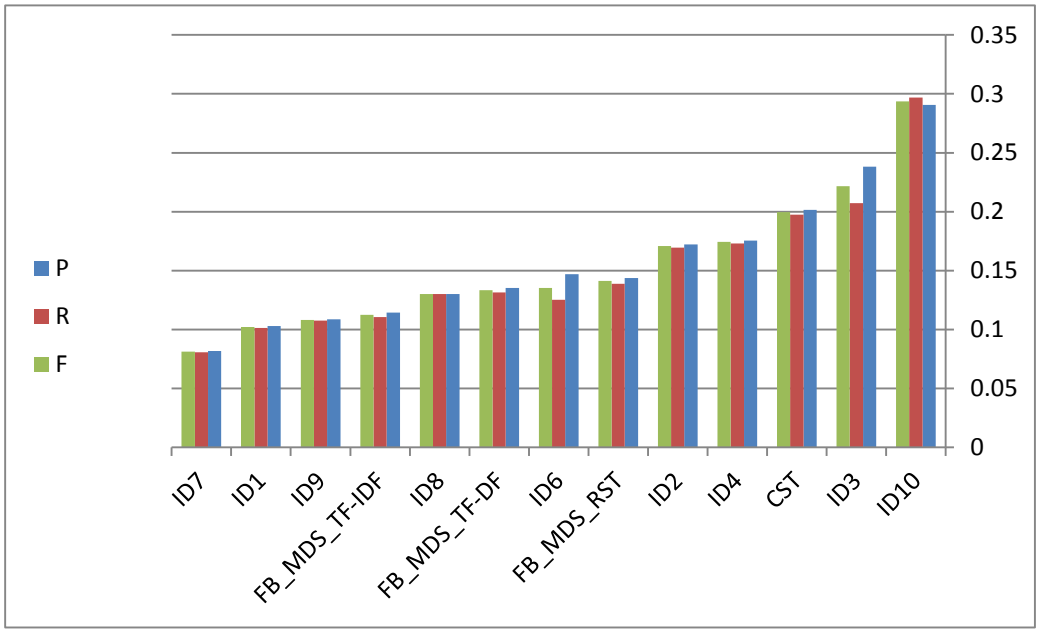

Figure 4.14: Arabic multi-document summarization ROUGE-2 results

For FB_MDS_TF-IDF, FB_MDS_TF-DF, and FB_MDS_RST methods, we note that ROUGE-1 result is better than the result of ROUGE-2. The reason for this is that the ROUGE-1 does not take into account the order of the words in the sentence for both the

peer and model summaries since it uses unigrams to compute the precision and recall values.

The two ROUGE results show a promising evaluation results for CST based method compared to the proposed systems and the peer summaries. For ROUGE-1 results, the CST based method has a better recall results than ID3 peer system, while ID3 has a good precision results which lead it to have a better F-measure score than CST based method. For ROUGE-2 results as described in Figure 4.14, we note that CST based method and ID3 system recall values are close to each other, while the ID3 precision value is better than CST precision value. Improvement are needed for CST precision and recall values to achieve a best summarization results.

Chapter 5

5. Conclusion and Future Works

5.1. Conclusion

In this thesis we proposed two methods for automatic Arabic multi-document summarization. The first method is based on feature extraction while the second method is based on Cross document structure theory (CST).

The first method was originally built on a previous research called AATSS uses features vector to score sentences and generate the final summary. AATSS is a system for single document summarization. In this thesis we adapt it for multi-document summarization. Three version were created which are FB_MDS_TF-IDF, FB_MDS_TF-DF, and FB_MDS_RST. In the first version we used TF-IDF as a term weighting while in the second one, we used TF-DF. In the last version, the documents are filtered using rhetorical structure theory to extract the important sentences, then apply feature based extraction to generate the final summary. FB_MDS_RST method shows a good results compared to FB_MDS_TF-IDF and FB_MDS_TF-DF.

The second method is based on CST. This method consists of three steps which are: CST training, relation identification, and summary generation. CST used to extract the highly relevant sentences to be included in the final summary. With the help of neural network classifier, the system automatically identifies the CST relations between sentences. After that, based on those relation we build a graph to link between the sentences. PageRank is used to score each sentence in the graph. After scoring the sentences, a novel redundancy removal step is applied on them. The final summary is generated based on the sentences scores by choosing the sentences that have the highest score to be included in the summary.

The performance of the proposed methods was evaluated using TAC 2011 MultiLing Pilot dataset. The CST based method was among the top three peers systems and shows better performance than feature based summarization method with precision and recall equals to 0.201 and 0.197 respectively.

5.2. Future Works

There are many areas can be improved, the following improvement were considered.

- Using different features combinations to identify the most important features that could be used to score the sentences.
- The CST dataset used in this thesis was created by translating another dataset. An improvement of our system could be creating an Arabic CST dataset annotated by Arabs writers. We believe that this will improve the process of CST classification.
- Improve the classification of CST relationships by exploring additional features and using different classifiers.
- Sentence re-ordering is a challenging problem. In this thesis we follow a simple approach for re-ordering, based on the publish date and CST relations.

REFERENCES

1. “Internet Users” [Online]. Available: http://www.internetlivestats.com/internet-users/. [Accessed April 24, 2015].

2. “Automatic summarization”, [Online]. http://en.wikipedia.org/w/index.php? title= Automatic_summarization. [Accessed April 1, 2015].

3. Ježek, Karel, and Josef Steinberger. "Automatic Text Summarization (The state of the art 2007 and new challenges)." In *Proceedings of Znalosti*, pp. 1-12. 2008.

4. Ding, Yuan. "A Survey on Multi-Document Summarization." *Department of Computer and Information Science University of Pennsylvania* (2004).

5. McKeown, Kathleen, Vasileios Hatzivassiloglou, Regina Barzilay, Barry Schiffman, David Evans, and Simone Teufel. "Columbia multi-document summarization: Approach and evaluation." (2001). In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval New Orleans, LA.

6. Barzilay, Regina, and Noemie Elhadad. "Inferring strategies for sentence ordering in multidocument news summarization." *Journal of Artificial Intelligence Research* (2002) 17: 35-55.

7. Donghong, Ji, and Nie Yu. "Sentence ordering based on cluster adjacency in multi-document summarization." In *The third international joint conference on natural language processing*, pp. 745-750. 2008.

8. Soudi, A., Neumann, G., & Van den Bosch, A. (2007). *Arabic computational morphology: knowledge-based and empirical methods* (pp.3-14). Springer Netherlands.

9. Roberts, Andrew, Latifa Al-Sulaiti, and Eric Atwell. "aConCorde: Towards an open-source, extendable concordancer for Arabic." *Corpora* 1, no. 1 (2006): 39-60.

10. Salton, Gerard, and Michael J. McGill. "Introduction to modern information retrieval." (2010).

11. Baeza-Yates, Ricardo, and Berthier Ribeiro-Neto. *Modern information retrieval*. Vol. 463. New York: ACM press, 1999.

12. Luhn, Hans Peter. "The automatic creation of literature abstracts." *IBM Journal of research and development* 2, no. 2 (1958): 159-165.

13. Baxendale, Phyllis B. "Machine-made index for technical literature: an experiment." *IBM Journal of Research and Development* 2, no. 4 (1958): 354-361.

14. Edmundson, Harold P. "New methods in automatic extracting." *Journal of the ACM (JACM)* 16, no. 2 (1969): 264-285.

15. Liu, Huan, Edward R. Dougherty, Jennifer G. Dy, Kari Torkkola, Eugene Tuv, Hanchuan Peng, Chris Ding et al. "Evolving feature selection." *Intelligent systems, IEEE* 20, no. 6 (2005): 64-76.

16. Khushaba, Rami N., Ahmed Al-Ani, and Adel Al-Jumaily. "Differential evolution based feature subset selection." In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1-4. IEEE, 2008.

17. Binwahlan, Mohammed Salem, Naomie Salim, and Ladda Suanmali. "Swarm based text summarization." In *Computer Science and Information Technology-Spring Conference, 2009. IACSITSC'09. International Association of*, pp. 145-150. IEEE, 2009.

18. Abuobieda, Albaraa, Naomie Salim, Ameer Tawfik Albaham, Ahmed Hamza Osman, and Yogan Jaya Kumar. "Text summarization features selection method using pseudo genetic-based model." In *Information Retrieval & Knowledge Management (CAMP), 2012 International Conference on*, pp. 193-197. IEEE, 2012.

19. Kwaik, Kathrein Abu. "Automatic Arabic Text Summarization System (AATSS) Based on Semantic Feature Extraction." Islamic University of Gaza, 2011.

20. “Arabic WordNet”, [Online]. http://globalwordnet.org/arabic-wordnet/, [Accessed February 13, 2015].

21. Sobh, Ibrahim Mohammed Abdul Hakim. "An optimized dual classification system for Arabic extractive generic text summarization." PhD diss., Faculty of Engineering, Cairo University, Giza, Egypt, 2009.

22. “Cluster analysis”, [Online]. http://en.wikipedia.org/wiki/Cluster_analysis [Accessed May 11, 2015].

23. Schlesinger, Judith D., Dianne P. O’leary, and John M. Conroy. "Arabic/English multi-document summarization with CLASSY—the past and the future." In*Computational Linguistics and Intelligent Text Processing*, pp. 568-581. Springer Berlin Heidelberg, 2008.

24. Radev, Dragomir R., Hongyan Jing, Małgorzata Styś, and Daniel Tam. "Centroid-based summarization of multiple documents." *Information Processing & Management* 40, no. 6 (2004): 919-938.

25. Sarkar, Kamal. "Sentence clustering-based summarization of multiple text documents." *Int. J. Comput. Sci. and Commun. Tech* 2, no. 1 (2009): 225-235.

26. Mihalcea, Rada, and Paul Tarau. "TextRank: Bringing order into texts." Association for Computational Linguistics, 2004.

27. Zhang, Junlin, Le Sun, and Quan Zhou. "A cue-based hub-authority approach for multi-document text summarization." Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference on. IEEE, 2005.

28. Thakkar, Khushboo S., Rajiv V. Dharaskar, and M. B. Chandak. "Graph-based algorithms for text summarization." In *Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on*, pp. 516-519. IEEE, 2010.

29. D. R. Radev, "A common theory of information fusion from multiple text sources step one: cross-document structure," presented at the Proceedings of the 1st SIGdial workshop on Discourse and dialogue - Volume 10, Hong Kong, 2000.

30. CastroJorge, M.L.d.R. and T.A.S. Pardo. “Experiments with CST-based multidocument summarization”. In Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing. 2010: Association for Computational Linguistics.

31. Almahy, Ibrahim, Naomie Salim, Yogan Jaya Kumar, and Ameer Tawfik. "Discussion Summarization Based On Cross-Document Relation Using Model Selection Technique." Advances in Neural Networks, Fuzzy Systems and Artificial Intelligence (2014): 218-229.

32. Kumar, Y.J., et al. “Multi-document summarization based on cross-document relation using voting technique”. In Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on. 2013: IEEE

33. Wan, X., Using only cross-document relationships for both generic and topic-focused multi- document summarizations. Information Retrieval, 2008. 11(1): p. 25-49.

34. Maziero, Erick Galani, and Thiago Alexandre Salgueiro Pardo. "Automatic Identification of Multi-document Relations." *Proceedings of the PROPOR 2012 PhD and MSc/MA Dissertation Contest* (2012): 1-8.

35. Miyabe, Yasunari, Hiroya Takamura, and Manabu Okumura. "Identifying Cross-Document Relations between Sentences." In *IJCNLP*, pp. 141-148. 2008.

36. Kumar, Yogan J., and Naomie Salim. "Automatic multi-document summarization approaches." Journal of Computer Science 8.1 (2011): 133-140.

37. Alotaiby, Fahad, Ibrahim Alkharashi, and Salah Foda. "Processing large Arabic text corpora: Preliminary analysis and results." In *Proceedings of the second international conference on Arabic language resources and tools*, pp. 78-82. 2009.

38. Althobaiti, Maha, Udo Kruschwitz, and Massimo Poesio. "AraNLP: A Java-based library for the processing of Arabic text." (2014).

39. Attia, Mohammed A. "Arabic tokenization system." In *Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources*, pp. 65-72. Association for Computational Linguistics, 2007.

40. Gharib, Tarek F., Mena B. Habib, and Zaki T. Fayed. "Arabic Text Classification Using Support Vector Machines." *IJ Comput. Appl.* 16, no. 4 (2009): 192-199.

41. Azmi, Aqil M., and Suha Al-Thanyyan. "A text summarizer for Arabic."*Computer Speech & Language* 26, no. 4 (2012): 260-273.

42. "Stemming", [Online]. http://en.wikipedia.org/wiki/Stemming, [Accessed May 14, 2015].

43. Al-Maimani, Maqbool R., A. A. Naamany, and Ahmed Zaki Abu Bakar. "Arabic information retrieval: techniques, tools and challenges." In *GCC Conference and Exhibition (GCC), 2011 IEEE*, pp. 541-544. IEEE, 2011.

44. Al Ameed, H., S. Al Ketbi, A. Al Kaabi, K. Al Shebli, N. Al Shamsi, N. Al Nuaimi, and S. Al Muhairi. "Arabic light stemmer: A new enhanced approach." In *The Second International Conference on Innovations in Information Technology (IIT'05)*, pp. 1-9. 2005.

45. Chen, Aitao, and Fredric C. Gey. "Building an Arabic Stemmer for Information Retrieval." In *TREC*, vol. 2002, pp. 631-639. 2002.

46. Kanaan, Ghassan, Riyad Al-Shalabi, M. Ababneh, and Alaa Al-Nobani. "Building an effective rule-based light stemmer for Arabic language to inprove search effectiveness." In *Innovations in Information Technology, 2008. IIT 2008. International Conference on*, pp. 312-316. IEEE, 2008.

47. Al Ameed, H., S. Al Ketbi, A. Al Kaabi, K. Al Shebli, N. Al Shamsi, N. Al Nuaimi, and S. Al Muhairi. "Arabic light stemmer: A new enhanced approach." In *The Second*

International Conference on Innovations in Information Technology (IIT'05), pp. 1-9. 2005.

48. L. Leah, B. Lisa and C. Margaret, "Light Stemming for Arabic Information Retrieval," Arabic Computational Morphology Text, Speech and Language Technology, vol. 38, pp. 221-243, 2007.

49. Dawoud, Hassan Mohammad. "Combining Different Approaches to Improve Arabic Text Documents Classification."

50. Nwesri, Abdusalam FA, Seyed MM Tahaghoghi, and Falk Scholer. "Stemming Arabic conjunctions and prepositions." In *String Processing and Information Retrieval*, pp. 206-217. Springer Berlin Heidelberg, 2005.

51. Kwaik, Kathrein Abu. Automatic Arabic Text Summarization System (AATSS) Based on Semantic Feature Extraction. Diss. Islamic University of Gaza, 2011.

52. Saad, Motaz K., and Wesam Ashour. "Arabic text classification using decision trees." In *Proceedings of the 12th international workshop on computer science and information technologies CSIT '2010, Moscow–Saint-Petersburg, Russia*. 2010.

53. Azara, Mohammed, Tamer Fatayer, and Alaa El-Halees. "Arabie text classification using Learning Vector Quantization." In *Informatics and Systems (INFOS), 2012 8th International Conference on*, pp. NLP-39. IEEE, 2012.

54. Salim, Naomie. "SRL-GSM: a hybrid approach based on semantic role labeling and general statistic method for text summarization." *Journal of Applied Sciences* 10, no. 3 (2010): 166-173.

55. Al-Hashemi, Rafeeq. "Text Summarization Extraction System (TSES) Using Extracted Keywords." *Int. Arab J. e-Technol.* 1, no. 4 (2010): 164-168.

56. Gupta, Vishal, and Gurpreet Singh Lehal. "Named Entity Recognition for Punjabi Language Text Summarization." *International Journal of Computer Applications* 33, no. 3 (2011): 28-32.

57. Hassel, Martin. "Exploitation of named entities in automatic text summarization for swedish." In *Proceedings of NODALIDA*, vol. 3. 2003.

58. Mann, William C., and Sandra A. Thompson. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute, 1987.

59. Mann, William C., and Sandra A. Thompson. "Rhetorical structure theory: Toward a functional theory of text organization." Text 8.3 (1988): 243-281.

60. Maziero, Erick Galani, Maria Lucía del Rosario Castro Jorge, and Thiago Alexandre Salgueiro Pardo. "Identifying Multidocument Relations." NLPCS 7 (2010): 60-69.

61. Radev, Dragomir and Otterbacher, Jahna and Zhang, and Zhu. "CSTBank: Cross-document Structure Theory Bank" http://tangra.si.umich.edu/clair/CSTBank (2003)

62. Cardoso, Paula CF, et al. "CSTNews-A Discourse-Annotated Corpus for Single and Multi-Document Summarization of News Texts in Brazilian Portuguese."the Proceedings of the 3rd RST Brazilian Meeting. 2011.

63. Althobaiti, Maha, Udo Kruschwitz, and Massimo Poesio. "AraNLP: A Java-based library for the processing of Arabic text." (2014).

64. "The Stanford Parser: A statistical parser", [Online]. http://nlp.stanford.edu/ software/lex-parser.shtml, [Accessed April 5, 2015].

65. Giannakopoulos, George, Mahmoud El-Haj, Benoît Favre, Marina Litvak, Josef Steinberger, and Vasudeva Varma. "TAC 2011 MultiLing pilot overview." (2011).

66. Giannakopoulos, George, and Vangelis Karkaletsis. "Summarization system evaluation variations based on n-gram graphs." (2010).

67. Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." Text Summarization Branches Out: Proceedings of the ACL-04 Workshop. 2004.