Exploiting Coverage, Coherence and Degree of Redundancy for An Extractive Multi-Document Summarization Mechanism *

Asma M. El-Saied , Nada A.D.

[1] Dept. of communication and electronics engineering, Mansoura High Institute of Engineering and technology, Mansoura, Egypt;
[2] Dept. of computers and systems, Faculty of engineering, Mansoura University, Mansoura, Egypt
Email: asma.m.alsaid@gmail.com
amsaieed@mc.edu.eg

ABSTRACT

Because of the expanding receptiveness about on the web information and the accessibility for many documents on the Internet, it becomes troublesome for a human to analyze and the review documents manually. This prompts trying the text mining strategies, automatic text summarization is one of the most significant text mining techniques. Many limitations are appeared in most of the current extractive multi-document summarization systems, such as low coherence among the summary sentences, low coverage, and high degree of redundancy. This paper provides an efficient framework for Automated Multi-Document Extractive Summarization (AM-DES).This framework introduce a new algorithm for measuring the Relatedness of the sentence. In addition to a new discriminative sentence selection method relies on sentence scoring and removing the redundant ones. An extensive experimental evaluation is conducted on three real data sets DUC2005, DUC2006 and DUC2007, indicating the importance of the proposed framework. Examining the effect of the proposed sentence Relatedness measure algorithm is provided to explore the effectiveness of the proposed AM-DES framework. The impact of this algorithm is shown by considering the semantic relations of the concept while calculating the semantic Relatedness. Evaluation metrics is used ROUGE-N, ROUGE_L as a case study and the results showed that the proposed AM-DES framework can result in a better summarization performance compared with the previous systems, where the generated summary is characterized by high coverage and cohesion.

Keywords : semantic sentence Relatedness, Feature Extraction, multi-document summarization, clustering

1 INTRODUCTION

With the recent growth in the amount of the documents available on the Internet, the powerful and fast automatic summarization has been more effective. The necessity to obtain maximum information in minimum time with least redundancy has led to more efforts to the field of summarization. Multi-document summarization intends to generate a summary that delivers most of the information content of a collection or set of documents. Automatic multi-document summarization has given a lot of interest in recent years, and it demonstrates practical application in the search systems and document management. Most of the existing studies are extraction-based methods [1,2]; they typically use a hierarchical model to select sentences from source text. However these methods suffer from a huge problem because of the highest ranked sentences usually produce redundant information. Furthermore, there are more aspects to consider when generating summaries such as Cohesion and coherence. Cohesion is aligned with the structure of the text surface level, granted as lexical and grammatical structures that interconnect text parts to each other using conjunctions, pronouns, time references and so on. The consistency of the semantic level structure of the document is hard to model, and needs further understandings of input text [4]. One of the main goals of automatic text summarization systems is to generate cohesive and coherent summaries. Classification of the automatic text summarization systems can be classified as extractive or abstractive according to the manner in which the final summary is generated. The extraction summary approach [5] aims to identify the most important concepts of the input document, and to give it as output. The abstract created for these methods may suffer from less coherency, but give a general sense of the content of the input document. In abstractive approaches [6], the system first understands the text, then briefly tells it in its own words. Furthermore, text summarization systems can be classified according to the number of documents as single-document summarization and multi-document summarization. In single document summarization, a shorter summary of a single document should be provided shorter than the original document, while in multi-document summarization a single

summary should be provided from two or more documents [7]. The aim of the summary can also set to classify of text summarization systems as generic summary or query-based one. In the generic summarization systems, the whole document is taken into consideration to generate a summary, in the query-based type, the produced summary is about the specific query only [8].

Development and research in automatic text summarization has been growing with the vast growth of online information services. The purpose of automatic text summarization is to take an input source text and present the most important content in a concise form in a manner sensitive to the needs of the user and the task. One of the hardest problems of Natural Language Processing (NLP) is summarization because, to do it properly, the point of a text must be well understood. This requires discourse processing, semantic analysis, and inferential interpretation (using world knowledge to group the content). The last procedure specifically is the most complicated, because systems simply cannot do it without a great deal of world knowledge. Therefore, attempts of performing correct abstraction have not been very practical so far. Luckily, extraction approximation is more feasible nowadays. A system needs to fluently indicate the most essential topics of the text, and return them to the user to create an extract. Although it won't be vitally coherent, but the user will be able to form an opinion of the overall content of the original document. Most automated systems today produce extracted summaries.

The major problems addressed in this paper are:

1) The low coverage in generating the final summary.
2) Inaccurate extraction of important sentences due to the leakage of semantic information.
3) A high degree of redundancy always in the extracted summaries.
4) Poor coherence among the selected sentences.

The main contributions of this paper are summarized as follows:

1) Constructing a new Graph-based document structure for more efficient document representation.
2) Proposing a new Semantic Relatedness Measure algorithm to overcome the problem of low coverage, that to help in clustering the sentences according to their Relatedness values.
3) Identifying the sentence main features to help in introducing a new scoring sentence method for more accurate selective sentences.
4) Combining the Maximal Marginal Relevance (MMR) technique with the sentence scoring method to detect the more relevant sentences with minimum Relatedness, and this reduce the degree of redundancy.
5) Implementing a sentence reordering approach to achieve a good coherence for the generated summary.

2 LITERATURE AND RELATED WORK

A summary is a condensed text containing the main ideas of the original content. The generation of a summary with a computer application is defined as automatic text summarization. Although it is an important topic of study today, only some software tools are available to the users and they are not commonly popular and this is because the low quality of the produced summaries. Generally, a lot of intelligence is hardly required for the creation of a good summary. Good understanding of a natural language, like many other NLP tasks, leads to a high quality automatic summarization. This is well known as an Artificial intelligence (AI) complete task, that is, it requires software with intelligence of a human that strong AI claims. Despite philosophical discussions about possibilities for strong AI, scientists could achieve good results in NLP tasks that are totally challenging, such as speech recognition, machine-based translation, domain specific question answering, etc. Although the problems won't be clearly solved, the results are very promising and effective. The motivation behind focusing on the research in the field is improving the quality of automatic summarization to this level of usefulness [9].

2.1 Some earlier approaches

Summarization is a research field with a long tradition. The first publications appeared in the 1950's and 1960's [12, 13], focusing on extractive strategies, while later work during the 1970's and 1980's took up trends in the field of AI and aimed for abstractive summarization [29]. The growing number and quality of natural language processing tools, such as robust part-of speech taggers and syntactic parsers, as well as the availability of suitable text corpora renewed interest in automatic summarization during the 1990's [29], with a shift back to extractive strategies. These years also saw the first applications of methods from Machine Learning (ML), and new research directions like multi-document summarization and multimedia summarization were being investigated.

Today, automatic summarization has become a vibrant field of research, with recent years seeing a rapid growth in publications. This growth has been fueled by the competitions conducted during the annual Document Understanding Conference (DUC) [29] and its successor, the Text Analysis Conference series (TAC), and the availability of summarization corpora that were created in the course of these competitions. Until recently, the attention of the research community focused on the tasks of generic and query-oriented multi-document summarization, typically of news material. However, this picture is changing 'rapidly, and many researchers are starting to investigate the summarization of non-news material (e.g. blogs or product reviews), or address other types of summarization such as update and opinion summarization. On the other hand, approaches which aim for abstractive summarization are still scarce and most systems opt for extractive strategies. Nevertheless, in recent research one can observe a tendency of using more complex linguistic processing during analysis and synthesis in order to move from simple passage extraction towards symbolic representations of source and summary content and reformulation for output generation [14].

The most popularly used recent extractive summarization methods are illustrated as follows:
In [43], a standard centroid-based method is introduced to rank sentences by calculating their salience using a set of features. This method extracts sentences resembling to three parameters (centroid value, first-sentence intersection and positional value). The centroid value of a sentence is evaluated as the average cosine Relatedness between the sentences and the rest of the sentences in the corpus. The intersection value is the cosine Relatedness between a sentence and the first sentence in the same document. The positional value is calculated as follows: the leading sentence is assigned score 1, and the score decreases by 1/n for each sentence, where n is the number of sentences in these documents. Then the three values are linearly combined with equal weights. In addition, a Graph-based method is used to construct sentences' graph, at which each node is a sentence in the document, and if there is a Relatedness between a pair of sentences, then there is an edge between this pair of sentences. In [23], the Lex Page Rank algorithm is proposed to compute the importance of the sentence in the graph structure. Other graph-based summarizations have been proposed by [44], [45] to adapt the sentence scoring by considering its edges that solve the problem of choosing the most important sentence.
Some methods firstly detect the important concepts from the documents using term-weighting methods such as TF-IDF [46], and then extract sentences that contain these words. Optimum ordering of the extracted sentences to create a coherent contextual sequence is a difficult problem. Ordering sentences extracted from a corpus into a coherent text is a non-trivial task [47]. In general, methods for sentence ordering in multi-document summarization can be classified into two approaches: making use of chronological information and learning the natural order of sentences from large corpus. The redundancy removal is the process of selecting one sentence of the extracted sentences that includes the same information. Several researches use sentence Relatedness in different ways to detect the duplicate information.

2.2 Problems formulation and plan of solution

Multi-document summarization is the task of producing a concise and fluent summary to deliver the major information for a given corpus. Multi-document summaries can be used for users to rapidly access document collections, and it also helps in information retrieval systems.
The existing multi-document summarization methods suffer from several limitations that need to be solved such as:
Interdependence: Most of the existing multi-document summarization methods work directly in the sentence space and many methods treat the sentences as independent of each other. Although few works tried to analyze the context or sequence information of the sentences, the document side knowledge, i.e. the topics embedded in the documents are ignored.
Coverage: Extraction process plays a vital role in the process of summarization. It presents important information that covers different subjects in the original documents. Many algorithms have been proposed to extract salient information from the original documents. The majority of these algorithms first identify important words from the source documents using term-weighting methods such as TF-IDF, and then extract phrases that contain these words. Various extractive summarization systems consider sentences selection as the final goal.
Inaccurate Selection: the sentence scores calculated from existing methods usually do not have very clear and rigorous probabilistic interpretations. Many if not all of the sentence scores are computed using various heuristics as few research efforts have been reported on using generative models for document summarization.
Redundancy: because the length is limited for an effective summary, and the existence of many extracted sentences that include the same information; it is preferable to select just one of them to be included in the summary. Many researchers use the Relatedness measure in different ways to identify the duplicate information.
Coherency: one of the problems that make the multi-document summarization differ from single document summarization is that it involves multiple sources of information that include the risk of higher redundant information than would typically be found in a single document. Besides, the organization and ordering of the extracted information from a set of documents to create a coherent summary is a non-trivial task.

3 THE PROPOSED AM-DES FRAMEWORK

As demonstrated in this document, the numbering for sections upper case Arabic numerals, then upper case Arabic numerals, separated by periods. Initial paragraphs after the section title are not indented. Only the initial, introductory paragraph has a drop cap.

The proposed AM-DES framework here intends to solve the limitations found in the previous extractive summarization systems such as:

- Low coverage in generating the final summary.
- Inaccurate extraction of important sentences.
- The degree of redundancy.
- Poor coherence among the selected sentences.

To overcome such limitations, it is necessary to construct a new graph-based document structure containing the most effective features of the sentences as well as applying a novel algorithm for sentence Relatedness measure. Moreover, there is a critical need for a discriminative sentence selection method afterwards removing the redundancy from the sentences, and finally reordering the generated summary sentences. To accomplish these necessities the proposed framework is divided into four major stages: Pre-processing stage, Interpreting stage, Extracting/Filtrating stage and Reordering/Generating stage.

The Pre-processing stage aims to analyze the text document; syntactical analysis and semantic analysis. The Interpreting stage represents the input document semantically using a graph-based structure through considering each sentence as a vertex in this graph along with edges corresponding to semantic relations between vertices. An Efficient Semantic Relatedness Measure (ESRM) algorithm is applied on this graph, and because it considers the sentence's relations while computing Relatedness, it overcomes the problem of low coverage in generating the final summary, and then sentences are clustered into some groups. In parallel with this process, the important sentence features are extracted, and the sentence vertex score is calculated according to these features. Those two processes are considered as an input for the extracting/filtrating stage which selects the most important sentences from each cluster with respect to their scores producing accurate selective sentences to form the candidate summary. A redundancy removal technique MMR is implemented to the candidate summary to get the more relevant sentences with minimum Relatedness and minimize the degree of redundancy in the formed summary. Finally, in the reordering/generating stage, the extracted sentences are reordered to generate the last form of a summary, which achieves a good coherence for the generated summary.

The components of the proposed framework are represented in Fig 1, where all the stages are depicted in a sequential manner.

This framework is divided into four major stages:

1) Pre-processing stage.
2) Interpreting stage.
3) Extracting/Filtrating stage.
4) Reordering/Generating stage.

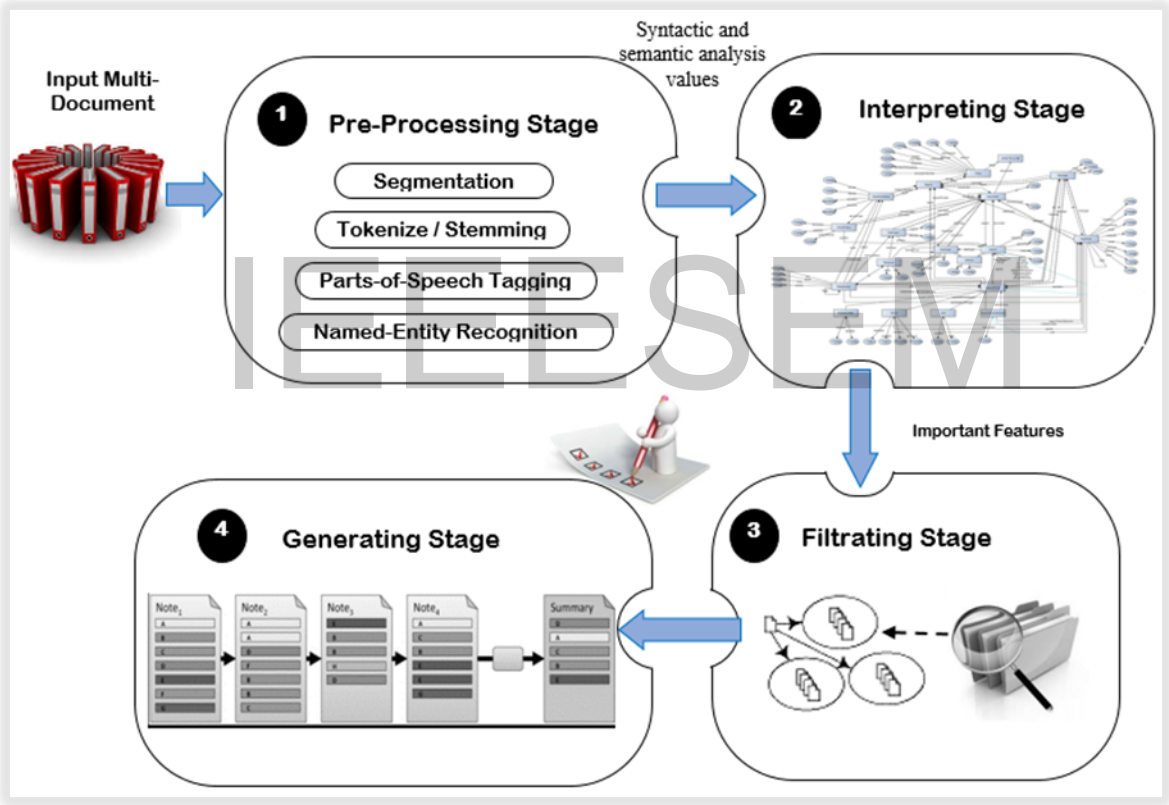

Fig 1 The proposed AE-MDS framework architecture

3.1 Pre-Processing Stage

The pre-processing stage is perhaps the most important stage in the area of computational linguistics, since the quality of the obtained summary depends on how efficient is the text represented. The pre-processing stage starts with shallow syntactic and semantic analysis of the input text, then extracts dependency relations and lexical relations (Synonyms, Is-A relation, and Part-of relation) for each word from WordNet [6, 7].

WordNet is an online lexical database system developed at Princeton University. In WordNet, nouns, verbs, adverbs and adjectives are organized by a variety of semantic relations into synonym sets (synsets), which represent one concept. Examples of semantic relations used by

WordNet are synonymy, autonomy, hyponymy, member, similar, domain, cause and so on. Relationship between the concepts such as hyponyms (i.e. more specific terms) is represented as semantic pointer linking the related concepts.
The Pre-processing stage of the AM-DES framework is responsible for accepting the input text, and converting it to pre-processed sentences. It consists of five main processes [48]: Sentence segmentation, tokenization, stemming, part-of-speech tagging and name entity recognition.

- Sentence Segmentation: The segmentation process identifies sentence boundaries.
- Tokenization: The tokenizer splits a plain text file into tokens. This includes, e.g., separating words and punctuation, identifying numbers, and so on.
- Stemming: The stemming obtains the root of words, so that the text processing is conducted on the roots and not on the original words. This allows relating more terms in the document. It is supposed that two words that have the same root represent the same concept. Basically, the process of stemming of the words is realized for reducing to a minimum common portion of a word called a stem. The stem is the portion of the word left after the removal of its affixes, prefixes and suffixes. Once implemented stemming, the document will contain only the roots of the words. This will simplify the representations of the documents.
- Parts-of-Speech (POS) Tagging: The POS tagger assigns to each word in the input sentence its proper part of speech such as nouns, verbs and determiners to reflect the word syntactic category; nouns, pronouns, prepositions, adverbs, articles, conjunctions, etc.
- Named-Entity Recognition (NER): The NER is also known as entity identification, which is a sub task of extracting information. It seeks to locate and classify atomic elements in text into predefined categories such as human names, organizations, locations, times, quantities, monetary values, percentages, etc.

The Stanford NLP toolkit[1] is used in this stage; it is an NLP software available to everyone. It provides statistical NLP, deep learning NLP, and rule-based NLP tools for major computational linguistics problems, which can be incorporated into applications with human language technology needs.

3.2 Interpreting Stage

The Interpreting stage starts with constructing a graph-based structure for each document by considering each sentence as a node in the graph. After that, it proceeds a clustering technique for the sentences (nodes) of the document using an efficient algorithm for semantic Relatedness measure. Then, it computes the sentence score by extracting the main features in each sentence. This helps in selecting the most important sentences that will form the summary in the next stage.

3. 2.1 Graph-based structure

It is the phase of representing each document in a group of vertices known as Document Graph (DG). Document sentences and the relations among them in the graph structure are illustrated in definition 1.

Definition1: Let Document Graph DG = (V, E) be a directed graph with a set of vertices V that represents the sentences, and set of edges E, where E is a subset of V * V which illustrates the semantic relations between vertices.

This DG annotated with:

1) Key: the key is the document identifier.
2) Category: the category is the topic that this document belongs to.
3) Title: the title is the document's title.
4) List of sentence vertices at which each vertex consists of:
   a) Sentence tree: a sentence tree (syntax tree) is an ordered, rooted tree that represents the syntactic structure of a sentence according to some context-free grammar. It is built using the Stanford NLP toolkit. The tree is used to ease the extraction of semantic relations of the main keywords of the sentence. More likely, it helps achieving coherence in the final selection process in the next stage.
   b) A list of sentence main features: sentence main features that are calculated.
   c) The sentence weight: the sentence weight is the summation of the weights of edges between this vertex and other vertices, divided by their number. This sentence weight is used to measure the centroid to identify sentences in each document that are central in the document.
   d) The sentence score: the sentence score is the summation of the main features.
   e) A list of sentence relations: the sentence relations which are taken into consideration are Synonyms, IS-A relation, Part-of relation.
5) A matrix of edges between the vertices: the matrix of edges represents the semantic Relatedness between each two vertices.

The constructed DG with a detailed description is illustrated in Fig 2.

3. 2.2 The proposed ESRM Algorithm

The Semantic Relatedness Measures play an increasingly important role in text summarization. Existing methods for computing sentence Relatedness have been adopted from approaches used for long text documents. These methods process sentences in a very high-dimensional space and are consequently inefficient. Traditionally, methods for detecting Relatedness between long documents have centered on analyzing shared words. Such methods are mostly effective when dealing with long documents because similar documents usually contain a degree of co-occurring words. However, in short documents, word co-occurrence may be rare or even null. This is mainly due to the inherent flexibility of natural language enabling people to express similar meanings using quite different sentences in terms of structure and word content.

[1] OpenNLP: http://opennlp.sourceforge.net/

Lin [49] calculates similarity by considering the information content (IC) of the Least Common Subsume (LCS) of two concepts c1 and c2, expressed by:

$$sim_L(c_1, c_2) = \frac{2 \times IC(lcs(c_1, c_2))}{IC(c_1) + IC(c_2)} \quad (1)$$

Where LCS is the most specific concept, which is a shared ancestor of the two concepts. The IC value is obtained by considering the negative log likelihood of encountering a concept in a given corpus; IC(c) = -log (P(c)). p(c) is the probability of encountering an instance of concept c; p(c) = freq(c)/n. The result is the ratio of the information shared in common to the total amount of information possessed by the two concepts.

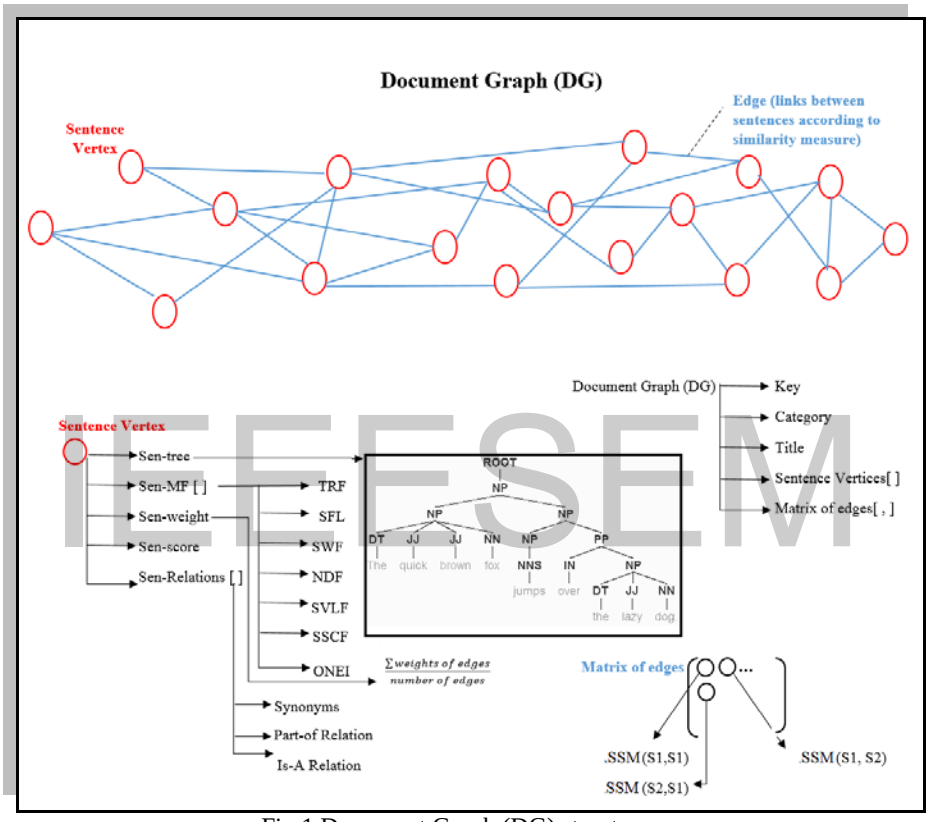

Fig 1 Document Graph (DG) structure

Traditionally, semantic similarity of Lin as illustrated in equation (1) is restricted to be used only for two concepts and do not include any relations about the sentence. For this purpose, the ESRM algorithm is proposed to represent the semantic similarity between two vertices by computing the semantic similarity between each concept in the Sentence Vertex (SV) with respect to the other SV's concepts. Both semantic and syntactic information play a vital role in conveying the meaning of sentences. As shown in Fig 3 the proposed ESRM algorithm considers the sentence's relations while computing Relatedness so it overcomes the problem of low coverage in generating the final summary.
The semantic Relatedness between two SVs is calculated as follows:

$$\text{ESRM}\,(SV_1, SV_2) = \frac{\sum_i^{SV_1 \cap SV_2} SWF_i * sim(SV_1, SV_2)}{\sum_i^{SV_1 \cup SV_2} SWF_i} \quad (2)$$

Where SV_1 is the vertex of the first sentence of the document, SV_2 is the vertex of the second sentence of the document. $SV_1 \cap SV_2$ are the shared elements, element's synonyms, element's is-A and element's part-of. SWFi is the frequency of element i and its synonyms and relations. $SV_1 \cup SV_2$ are all elements, element's synonyms, element's is-A and element's part-of in both SV_1 and SV_2. While sim(SV_1,SV_2) is calculated using the IC equation as illustrated in formula (3).

The semantic Relatedness (sim(SV_1,SV_2)) between two vertices is calculated by computing the semantic Relatedness between each concept in the vertex with respect to the other vertex's concepts.

$$sim(SV_1, SV_2) = \sum_{i}^{n} \sum_{j}^{m} \frac{2 \times \log\left(f\left(lcs\left(SV_1(i), SV_2(j)\right)\right)\right)}{\log\left(\frac{f(SV_1(i))}{n}\right) + \log\left(\frac{f(SV_2(j))}{m}\right)} \quad (3)$$

Where i is the concepts' index in vertex SV_1, j is the concepts' index in vertex SV_2, n number of concepts in vertex SV_1, and m is the number of concepts in the vertex SV_2. f(lcs(SV_1(i),SV_2(j))) is the frequency of the shared concepts between SV_1 and SV_2. f(SV_1(i)) is the frequency of the concepts in SV_1 at the document level, f(SV_2(j)) is the frequency of the concepts in SV_2 at the document level.

3. 2.3 Sentence Clustering

Hierarchical Agglomerative Clustering (HAC) algorithm might be the most commonly used algorithm among numerous document-clustering algorithms [41]. For the HAC algorithm, an open source library is used, i.e., the C clustering library [47]. HAC is a straightforward greedy algorithm that produces a hierarchical grouping of the data. It starts with all instances each in its own cluster, and then repeatedly merges the two clusters that are most similar at each iteration. There are different approaches of how to find the Relatedness between two clusters [50].

The accuracy of clustering approach is determined based on Relatedness or distance measures. A variety of Relatedness measures have been proposed so far [51]. To improve the accuracy of document clustering, the proposed ESRM algorithm is used. It plays a vital role in clustering the documents.

```
Input: List of n sentence vertices in DG
Output: The matrix of edges ME(n,n) values
For i=1 to n in ME
For j=1 to m in ME
ME (i, j) = ESSSM (SVi, SVj)
End
End
Function ESSSM (SV1, SV2)
Foreach concept i=1 to n in SV1
Foreach concept j=1 to m in SV2
LCS= compute LCS (SV1 (i), SV2 (j))
FLCS= frequency (LCS)
F1=frequency (SV1 (i))
F2= frequency (SV2 (j))
Sim+= 2* log (FLCS)/ (log (F_1/n) + log (F_2/m))
End
End
Shared= getshared_element (SV1, SV2)
All= getall_element (SV1, SV2)
Foreach element in shared
SWF_shared+=frequency (element)
End
Foreach element in all
SWF_all+=frequency (element)
End
ESSSM= (SWF_shared * sim)/SWF_all
Return ESSSM
End function
```


Fig 3 The proposed ESSSM algorithm

Given a set of N vertices to be clustered, and an N*N Relatedness matrix (matrix of edges), for each document graph DG the hierarchical clustering algorithm is:

1) Initialize each vertex as a cluster, so that if you have N vertices, you now have N clusters, each containing just one vertex. Let the similarities between the clusters equal the similarities between the vertices they contain.
2) Merge the pair with the highest Relatedness to each other.
3) Compute similarities between the new cluster and each of the old clusters.
4) Repeat steps 2 and 3 until all vertices are clustered into the required number of clusters.

3. 2.4 Feature Extraction

Sentences are given importance scores, and this acts as a goodness measure for the sentence. Each sentence is represented by a set of features, and the score is a function of the weighted sum of the individual feature values.

For each SV in the document, a sentence score will be calculated based on the combination of the sentence's features score. Each feature

score can have a value between 0 and 1. Some of these features are presented in the previous studies [30 ,38,52,53], and some of them are enhanced and added as shown in the feature extraction algorithm Fig 4. These features are extracted in the interpreting stage as follows:

a) Title Resemblance feature (TRF)

Sentences containing concepts that appear in the title are indicative of the document [30]. These sentences have greater chances for being included in the summary. This feature score is calculated as follows for SV:

$$TRF(SV) = \frac{concepts\ in\ SV\ \cap\ concepts\ in\ the\ title}{concepts\ in\ SV\ \cup concepts\ in\ the\ title} \quad (4)$$

b) Sentence weight feature (SWF)

This feature calculates the sentence weight score based on Term, Synonyms and Relations Frequencies - Inverse Document Frequency (TSRF-IDF) value for each term in a sentence vertex and takes their average. The proposed SWF takes into account not only the frequency of a term but also the frequencies of term's synonyms and term's relations. So the TSRF-IDF score for a term t in the document d in a given corpus is calculated as follows:

$$TSRFIDF(t,s,r,d) = TF(t,d) * log\frac{n}{df(t,s,r)} \quad (5)$$

Where TF (t,d) is the frequency of the term (t), the frequency of the term's synonym (s) and the frequency of the term's relations (r) at the document level (d). The value of n is the total number of documents in the corpus, df (t, s, r) is the number of documents in which term t, its synonyms s and its relations r occur.

The feature score for a sentence vertex SV is the average of the TSRF-IDF scores of all the terms in SV.

$$SWF(SV) = \sum_{i}^{m} TSRF - IDF(t_i,s,r,d)) \quad (6)$$

Where i is the index of the SV's terms. m is the total number of terms in the SV. IDF is the Inverse Document Frequency to measure the general importance of the term t in a corpus of documents. IDF is performed by dividing the number of all documents by the number of documents containing this term (t) [52].

c) Numeric data feature (NDF)

Usually the sentence that contains numerical data is an important one and it is most probably being included in the document summary [53]. The score for this feature is calculated as follows for SV:

$$NDF(SV) = \frac{number\ of\ concepts\ which\ contain\ numeric\ data\ in\ SV}{total\ number\ of\ concepts\ in\ SV} \quad (7)$$

d) Sentence vertex links feature (SVLF)

The sentence link feature is defined as the number of links connecting the SV to other vertices on the graph [53].

$$SVLF(SV) = \frac{number\ of\ SV\ links}{total\ number\ of\ links\ in\ the\ doc} \quad (8)$$

Where the SV links are the corresponding values of the SV in the matrix of edges, and the total number of links in the document is the number of all values in the matrix of edges.

e) Sentence-to-Sentence Cohesion feature (SSCF)

This feature is obtained based on the graph structure previously constructed. Each row in the matrix of edges represents the semantic Relatedness between each SV and all other vertices on the graph. Sentence-to-sentence cohesion score is calculated by adding up those Relatedness value. Scores with greater values indicate sentences with larger cohesion. This feature score is calculated as follows for SV:

$$SSCF(SV_i,D) = \sum_{j}^{n} SWF_j * ASSM(SV_i,SV_j) \quad (9)$$

Where i is the index of the specific SV. n is the number of vertices in the Document D. SWF is the previously calculated sentence weight feature. ESRM (SVi,SVj) is the value of the proposed ESRM algorithm between the specific SVi, and all other vertices in the graph SVj.

f) Occurrence of Non-Essential Information (ONEI)

Considering that some words are indicators of non-essential information that can be referred to as speech markers, examples of these words are, "because", "furthermore" and "additionally". They typically occur in the beginning of a sentence. This feature is calculated as follows for SV:

$$ONEI(SV) = \frac{number\ of\ non\ essential\ concepts\ in\ SV}{total\ number\ of\ concepts\ in\ SV} \quad (10)$$

Input: document graph structure (DG)
Output: List of features for each sentence vertex sv
Function feature_extraction()
Foreeach(vertex sv in DG)
TRF(SV) =concepts in SV ∩ concepts in the title/concepts in SV ∪concepts in the title
// to detect sentences containing concepts that appear in the title
$SWF(SV) = \sum_{j}^{m} TSRF - IDF(t_j, s, r, d))$
// to calculate the sentence weight score based on Term, Synonyms and Relations Frequencies - Inverse Document Frequency
$TSRFIDF(t, s, r, d) = TF(t, d) * \log[n/df(t,s,r)]$
// Term, Synonyms and Relations Frequencies - Inverse Document Frequency
NDF(SV) =number of concepts which contain numeric data in SV/ total number of concepts in SV
// to detect sentences contain numerical data
SVLF(SV) =number of SV links/total number of links in the doc
// the number of links connecting the SV to other vertices on the graph
$SSCF(SV_i, D) = \sum_{j}^{n} SWF_j * SSM(SV_i, SV_j)$
// Sentence-to-sentence cohesion score is calculated by adding up the similarity values (ESSSM).
ONEI(SV) =number of non essential concepts in SV/total number of concepts in SV
//to detect the sentences with non-essential information
SV.feature .add (TRF, SWF, NDF, SVLF, SSCF, ONEI)
End
Return SV.feature
End function

Fig 4 Feature extraction algorithm

3.2.5 Sentence Scoring

After identifying the sentence features, each sentence is assigned a score which indicates its importance. The next step is to combine the value of the sentence features computed in the previous section to score the sentences. The scores can be used to order sentences and pick the most important ones. The probability of a sentence to be present in the summary is proportional to its score. Each sentence is now represented by the six features, and the overall sentence score is computed as follows:

$$Sen\_score(SV) = TRF + SWF + NDF + SVLF + SSCF + ONEI \quad (11)$$

3. 3 Filtrating stage

This stage aims to extract a candidate summary through a pre-selection process, based on the output of sentence clustering and sentence scoring; to detect the most significant ones. As well, filtrating this candidate summary is implemented through the final selection process, which is based on the MMR technique for reducing sentences' redundancy.

3. 3.1 Pre-selection process

The pre-selection process starts with choosing the most significant sentences for the candidate summary that based on the sentence clustering and the sentence scoring as follows:

1) The obtained sentence scores produce a ranked list of sentences with the highest scores (top n scores).
2) The obtained clusters are used to group the most similar sentences in m classes.
3) The overlap between n selected sentences and m classes includes the most important sentences in the document. For each cluster in document Di, Sentences are checked. The sentences with the highest scores in the list are chosen, forming Xi for this document.

Then the candidate summary denoted as X to be a sequence of sentences (X1, X2, …, Xn) as shown in Fig 5.

3. 3.2 Final selection process

Due to length limitations required for an effective summary, and the existence of many extracted sentences that include the same information which reduce the summary readability and increase the degree of redundancy; it is desirable to select just one of them to include in the summary. The MMR technique is used to select sentences by calculating the semantic Relatedness between a sentence and the document topic

and also the sentence and previously selected sentences as in formula 12. MMR aims to choose relevant sentences and dislodges redundant ones [54].

$$MMR(SV_i) = Arg\ max[\lambda * sim(SV_i, D) - (1 - \lambda) * sim(SV_i, Summ)] \qquad (12)$$

Where D is the document vector, Summ represents the sentences that have been extracted into the summary, and λ is used to adjust the combined score to emphasize the relevance or to avoid redundancy. The Relatedness functions $Sim(SV_i,D)$ and $Sim(SV_i,Summ)$ represent the Relatedness of a sentence to the entire document and to the selected summary, respectively. The sentences with the highest MMR scores will be repeatedly chosen into the summary until the summary reaches a predefined proper size.

In the final selection process, the redundancy removal technique is applied to X sentences of each document extracted from the pre-selection process. The MMR is adapted for fitting the summarization system; it extracts important sentences with taking the coherence into account. The MMR formula is used, but with the Relatedness values previously calculated in this .

$$MMR = Arg\ max[\lambda * SSCF(SV_i, D) - (1 - \lambda) * SSCF(SV_i, Summ)] \qquad (13)$$

Where SSCF(SVi,D) and SSCF(SVi,Summ) represent the sentence to sentence cohesion feature score of SV to the entire document and to the candidate summary, respectively.

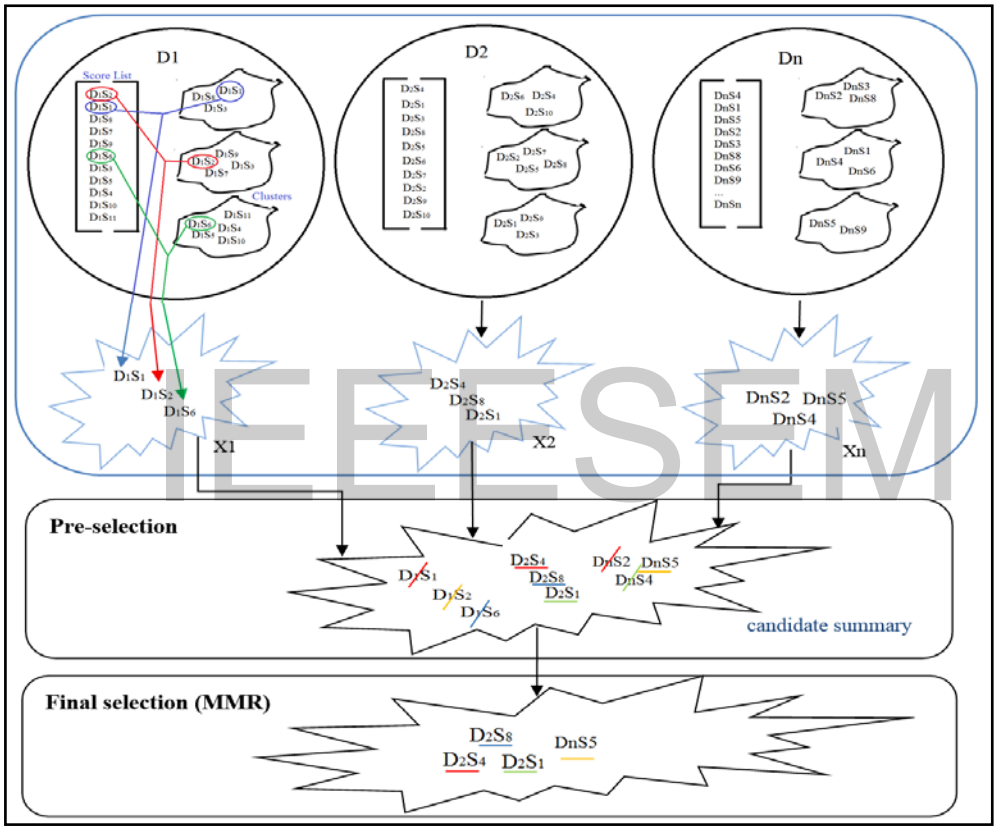

Fig 5 Extracting/ Filtrating stage

The terms of this equation are considered as the term for extracting the relevant sentences and the term for eliminating redundancy, respectively. The value of parameter λ coordinates those two effects. The parameter λ has the range [0-1]. The closer to 0 it is, the more effective the elimination of redundancy is. Since the adequate value of λ may depend on the target set of documents, the value should be selected adequately. The final summary algorithm is proposed to calculate the two processes of the extracting/filtrating stage as shown in Fig 6.

3. 4 Generating stage

Most of the existing summarization systems use sentence or paragraph extraction, which finds significant textual segments in the original documents, and compiles them in a summary form. After selecting significant sentences as a material for the summary, a proper arrangement for these sentences must be fulfilled, afterwards editing each sentence by deleting unnecessary parts or inserting necessary expressions. In this stage, a chronological approach [55] to perform a coherent text structure for summarizing documents or newspaper articles is used.

When there are sentences having the same time stamp, sentence position and sentence connectivity take place. Original ordering is restored if two sentences have the same time stamp and belong to the same article. And if sentences have the same time stamp but they do not belong to the same article, the Relatedness between these sentences is checked; sentence with higher Relatedness with the previously ordered sentences is taken to assure sentence connectivity.

```
Input: DG with list of SV & matrix ME
Output: final summary with x sentence.
Function generating_summary()
  Foreach SV in DG
Compute score=  SV.TRF+SV.SWF +SV.NDF +SV.SVLF+SV.SSCF+SV.ONEI
  End
Get n sv with highest score
Obtain m classes using HAC
Candidate_summary= overlap(n,m)
// reduce degree of redundancy
λ=0.7
  Foreach SV in DG
MMR (SV) = max (λ* SSCF (SV, DG)-(1-λ)*SSCF (SV, candidate_summary))
  End
Final_summary= select_highest(SV.MMR)
End function
```

Fig 6 Final summary

4 EXPERIMENTS AND EVALUATION

The experimental evaluation for a summary is a difficult task because there isn't an ideal summary for a given document or corpus. The preceding evaluations have found that the agreement between human summarizers is quite low, both for evaluating and generating summaries. Besides, manual evaluation is too expensive as stated by [56] large-scale manual evaluation of summaries as in the DUC conferences would require over 3000 hours of human efforts. Hence, an evaluation metric having high correlation with human scores would obviate the process of manual evaluation. As the Relatedness measure is one of the most important core components in summarization, it is experimented with various Relatedness measures.

Evaluation measures are categorized in sub-categories in [43, 57], which can be seen in Fig 7 Text quality based evaluation is done by human annotators who give score to each summary according to a predefined scale. Content based evaluation is done against a grand-truth summary, which is created by a human. Content based evaluations can use information of matching sentences (co-selection based evaluation) or matching words (content based evaluation). Task based evaluations measure the quality of the summary for a given task, e.g. question answering.

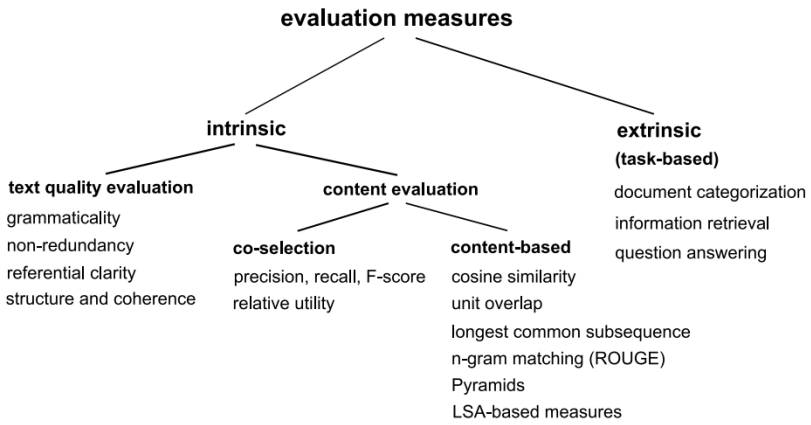

Fig 7 The taxonomy of summary evaluation

4.1 Evaluation metrics

In order to evaluate the quality of the generated summaries by different methods, there are several metrics to evaluate the proposed ESRM and others for the proposed EFFICIENT AM-DES . The extracted summaries must be compared with the ideal summaries to check if they are powerful enough or not. This comparison is based on evaluating the selected sentences to validate the implementation of the proposed EFFICIENT AM-DES framework, examining the effect of the proposed ESRM algorithm on the automatic summarization process.

Precision is defined in the glossary of [58] as “an information retrieval performance measure that quantifies the fraction of retrieved documents which are known to be relevant.” For text summarization, it is the division of extracted summary sentences and ideal summary sentences intersection over whole extracted summary sentences.

Recall is defined as “an information retrieval performance measure that quantifies the fraction of known relevant documents which were ef-

fectively retrieved", in the glossary of [58]. From the point of view of document summarization, it is the division of extracted summary sentences and ideal summary sentences intersection over the ideal summary sentences.

		Human judgment	
		true	false
System judgment	true	TP	FP
	false	FN	TN

F-score (F-measure) is a statistical measure that combines both precision and recall. Traditionally it is defined as the harmonic mean of precision and recall. F-score values changes in the interval of 0 and 1, where best result is 1.
The evaluation metric for the proposed ESRM is identified by measuring precision, recall, and f-measure values. They are defined as follows [59]:

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$Precesion = \frac{TP}{TP + FP} \tag{15}$$

$$Fmeasure = \frac{2 * Recall * Precesion}{Recall + Precesion} \tag{16}$$

TP stands for the number of pairs correctly similar. TN stands for the number of pairs correctly non-similar. FP stands for the number of pairs incorrectly similar. And FN stands for the number of pairs incorrectly non-similar. Recall is defined as the number of true positives divided by the total number of pairs that actually belong to the positive class. Precision is the number of true positives divided by the total number of pairs labeled as belonging to the positive class, while F-measure is the geometric mean of recall and precision.
The relatedness or similarity measures are inherited from probability theory and known as the correlation coefficient [60]. The correlation coefficient is one of the most widely used measures to describe the relatedness r between two vectors, X and Y. The correlation coefficient is one of the most widely used measures to describe the relatedness r between two vectors, X and Y. The correlation coefficient is a relatively efficient relatedness measure, which is a symmetrical measure of the linear dependence between two random variables. Therefore, the correlation coefficient can be considered the coefficient for the linear relationship between corresponding values of X and Y. The correlation coefficient r between sequences X = {xi: i = 1, . . ., n} and Y = {yi: i = 1, . . ., n} is defined by:

$$r = \frac{\sum_{i=1}^{n} X_i Y_i}{\sqrt{(\sum_{i=1}^{n} X_i^2) * (\sum_{i=1}^{n} Y_i^2)}} \tag{17}$$

While the evaluation metric for the proposed EFFICIENT AM-DES framework is the Recall-Oriented Understudy for Gisting Evaluation ROUGE [56] evaluation toolkit, which is adopted by DUC for automatic summarization evaluation. It measures summary quality by counting overlapping units such as the n-gram, word sequences and word pairs between the candidate summary and the reference summary. Several automatic evaluation methods [61], are implemented in ROUGE, such as ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-SU. ROUGE-N is an n-gram computed as follows:

$$Rouge - N_{recall} = \frac{\sum S \in ref \sum gram_n \in S\, Count_{match}(gram_n)}{\sum S \in ref \sum gram_n \in S\, Count(gram_n)} \tag{18}$$

$$Rouge - N_{precision} = \frac{\sum S \in ref \sum gram_n \in S\, Count_{match}(gram_n)}{\sum S \in ref \sum gram_n \in S\, Count(gram_n)} \tag{19}$$

Where n is the length of the n-gram, and ref stands for the set of the reference summaries. Count match (gramn) is the maximum number of n-grams co-occurring in a candidate summary and the reference summaries, and Count(gramn) is the number of n-grams in the reference summaries. ROUGE-L uses the longest common subsequence (LCS) statistics, while ROUGE-W is based on weighted LCS, and ROUGE-SU is based on skip-bigram plus unigram. Each of these evaluation methods in ROUGE can generate three scores (recall, precision, and F-measure). Only the average F-measure scores generated by ROUGE-1, ROUGE-2, and ROUGE-L are reported, to compare the proposed AM-DES framework to other implemented systems. Intuitively, the higher the ROUGE scores, the more similar the two summaries are.

4.2 Data sets

An extensive experimental evaluation is conducted on real data sets from various domains, showing the efficiency of the proposed AM-DES framework. Some of these experiments revealed some interesting trends in terms of selecting the important sentences based on the proposed

ESRM algorithm. Clearly, the results show the effectiveness of the proposed ESRM algorithm for the summarization process. The validation of this is accomplished on public data set Microsoft Research Paraphrase (MSRP)[2] corpus. The MSRP consists of 1,725 test pairs and 4,076 training pairs. The pairs were automatically collected from thousands of news sources. Then subsequently labeled by two human annotators who determined whether the two sentences in a pair were semantically equivalent or not.

On the other hand, Document Understanding Conference (DUC)[3] has organized yearly evaluation of document summarization. The standard summarization benchmark DUC2005, DUC2006 and DUC2007 data sets are used for validating the proposed AM-DES framework. DUC 2005 used in these experiments is partitioned into 50 topic sets, each containing 25–50 documents. DUC2006 contains 50 document sets while DUC2007 contains 45 document sets. Every document set in DUC2006, and DUC2007 has 25 news articles. Each document set consists of several articles written by various authors, which is also the ground truth of the evaluation. Every sentence is either used in its entirety or not at all for constructing a summary. The length of a result summary is limited by 250 tokens [61].

4.3 Experimental Results

In this section, the results of the experiments are analyzed in details. Two experiments are done using the evaluation data sets outlined in the previous section. Experiment 1are used to validate the similarity, and they prove that the proposed ESRM algorithm performs the best results among the compared methods. As well, Experiment 2 is used to validate the effectiveness of the proposed AM-DES framework. The efficiency is achieved through selecting the most relevant sentences with the least redundancy.

4.3.1 Experiment 1

In MSRP, the proposed ESRM determines the number of correctly identified paraphrase pairs in the corpus and compares the result with STS approach [59] and LG approach [64]. The effectiveness of the proposed ESRM algorithm is measured by two quantities and one combined measure, named “recall” and “precision” rate. Fig 8 depicts the precision, recall, and F-measure versus similarity threshold practical values.

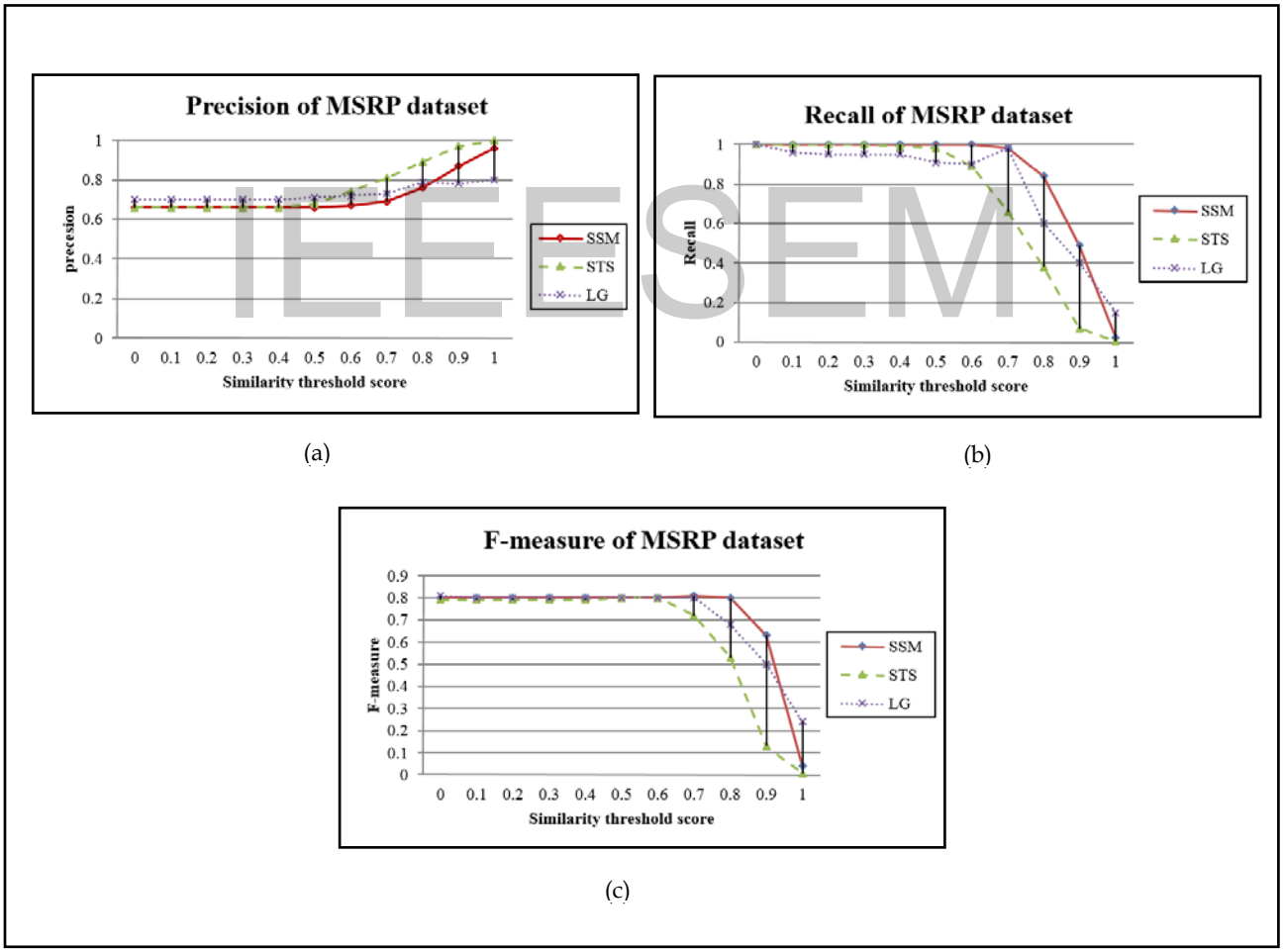

Figure 8 Comparison results on MSRP data set

[2] http://research.microsoft.com/en-us/downloads/607d14d9-20cd-47e3-85bc-a2f65cd28042/
[3] http://duc.nist.gov/data.html

A comparison between the proposed ESRM algorithm, STS approach [59], and grammar based approach LG [64] is illustrated in these figures. The Precision values are satisfactory but not the best, compared to STS approach and LG approach. This is because the proposed ESRM algorithm increases the number of true-positive TP but can't decrease the number of false-positive FP. However, the recall values achieve the best results due to increasing the number of true-positive TP and decreasing the number of false-negative FN. Subsequently the performance of the f-measure increases; it is a combination between them. The result shows that the proposed ESRM algorithm outperforms the result of STS approach [59] and LG approach [64] with 0.7 to 1.0. In details, the proposed ESRM improves the effectiveness in terms of f-measure with a ratio 9% greater than STS at threshold 0.8, 27% greater than STS and 12% greater than LG approach at threshold 0.8, and 50 % greater than STS and 13% greater than LG approach at threshold 0.9.

4.2.4 Experiment 2

This experiment shows the necessity of studying the impact of the proposed ESRM algorithm on the AM-DES framework. This impact is achieved through filtering the most relevant with least redundant, so the results would be comparable with other previous studies in the same field. With comparison to the average ROUGE values for other methods, the proposed AM-DES framework achieves significant improvement.

The proposed AM-DES is compared with several state-of-the-art text extraction methods described briefly as follows:

1) Random: The method selects sentences randomly for each document collection [61].
2) LSA: The method performs latent semantic analysis on terms by sentences' matrix to select sentences having the greatest combined weights across all important topics [65].
3) Document Summarization based on Data Reconstruction (DSDR) [61] represents each sentence as a non-negative linear combination of the summary sentences. And it uses sparse coding to select the summary sentences.

Results of comparison reported in Tables 1, 2 and 3. It is observed that the proposed AM-DES demonstrates the best ROUGE values and outperforms all the other systems on the three data sets. Among other methods the best results have been shown by the DSDR [61] method on DUC2006 and DUC2007 data sets, respectively. The comparison with the method DSDR [61] on DUC2006 data set shows that the proposed AM-DES improves the performance by 2%, 1.4% and 3.2% in terms ROUGE-1, ROUGE-2 and ROUGE-L metrics, respectively. Comparison also with the LSA [65] on DUC2007 data set shows the proposed AM-DES improves the performance by 12.1%,6.1% and 16.6% in terms ROUGE-1, ROUGE-2 and ROUGE-L metrics, respectively. Moreover, Comparison with the Random method on DUC2005 data set shows the proposed AM-DES improves the performance by 13%,1.5% and 14.5% in terms ROUGE-1, ROUGE-2 and ROUGE-L metrics, respectively. The experimental results provide strong evidence that the proposed AM-DES framework is a viable method for automatic document summarization.

TABLE 1
OVERALL PERFORMANCE COMPARISON ON DUC 2006 USING ROUGE-1, ROUGE-2, AND ROUGE-L.

System	Rouge-1	Rouge-2	Rouge-L
Random	0.285	0.042	0.259
LSA	0.2578	0.037	0.232
DSDR	0.33	0.06	0.298
AM-DES	0.35	0.074	0.33

TABLE 2
OVERALL PERFORMANCE COMPARISON ON DUC 2007 USING ROUGE-1, ROUGE-2, AND ROUGE-L.

System	Rouge-1	Rouge-2	Rouge-L
Random	0.32	0.054	0.29
LSA	0.259	0.036	0.22
DSDR	0.39	0.074	0.35
AM-DES	0.38	0.097	0.386

TABLE 3
OVERALL PERFORMANCE COMPARISON ON DUC 2005 USING ROUGE-1, ROUGE-2, AND ROUGE-L.

System	Rouge-1	Rouge-2	Rouge-L
Random	0.31	0.063	0.345
LSA	0.34	0.065	0.349
AM-DES	0.44	0.078	0.49

5 CONCLUSION ANF FUTURE WORK

This paper proves that the automatic text summarization can be improved by considering the semantic relations while extracting the summary. Thus, a four-stage framework was proposed for generating a salient and concise summary. The paper introduces an efficient AM-DES framework to solve the limitations found in the previous extractive summarization systems such as: Low coverage in generating the final summary, inaccurate extraction of important sentences, the degree of redundancy, and poor coherence among the selected sentences. This framework improves the effectiveness of the automatic text summarization process of the textual documents. First, the document is transformed into a graph structure form to be clustered using HAC algorithm after computing the semantic Relatedness of each sentence using an ESRM algorithm. Besides, the sentences are given specific scores based on a new feature extraction algorithm, then, the sentences are selected and subjected to a redundancy removal technique. Finally, they are reordered to generate a coherent summary.
Some possible ways and ideas to extend this work in the future are indicated: The work can be extended by adding new features and comparing the results, which, in turn, should improve the quality of the summaries. Several major summarization subtasks such as sentence reduction and sentence realization were not implemented in the system. These are very interesting and constitute a challenging field of research. Many challenges are left to be investigated in sentence compression using syntactic pruning techniques to approximate what humans do in sentence compression.

REFERENCES

[1] Yadav, Chandra Shekhar, and Aditi Sharan. "Hybrid approach for single text document summarization using statistical and sentiment features."International Journal of Information Retrieval Research (IJIRR) 5.4: 46-70, 2015.
[2] Nenkova, Ani, and Kathleen McKeown. "A survey of text summarization techniques." Mining Text Data. Springer US, 43-76, 2012.
[3] Li, Liangda, et al. "Enhancing diversity, coverage and balance for summarization through structure learning." Proceedings of the 18th international conference on World wide web. ACM, 2009.
[4] Jaya Kumar, Yogan. "Automatic multi document summarization approaches." Journal of Computer Science 8.1, 133-140, 2012.
[5] Carenini, Giuseppe, and Jackie Chi Kit Cheung. "Extractive vs. NLG-based abstractive summarization of evaluative text: The effect of corpus controversiality." Proceedings of the Fifth International Natural Language Generation Conference. Association for Computational Linguistics, 2008.
[6] Genest, Pierre-Etienne, and Guy Lapalme. "Framework for abstractive summarization using text-to-text generation." Proceedings of the Workshop on Monolingual Text-To-Text Generation. Association for Computational Linguistics, 2011.
[7] Kogilavani, A., and P. Balasubramani. "Clustering and feature specific sentence extraction based summarization of multiple documents." International Journal of Computer Science Information Technology 2.4: 99-111, 2010.
[8] Murali, R. V. V., SY Pavan Kumar, and Satyananda Reddy. "A hybrid method for query based automatic summarization system." International Journal of Computer Applications 68.6, 2013.
[9] Gleb Sizov, "Extraction-Based Automatic Summarization Theoretical and Empirical Investigation of Summarization Techniques", Department of Computer and Information Science, Norwegian University of Science and Technology, 2010.
[10] Radev, D. R., Hovy, E., & McKeown, K., "Introduction to the special issue on summarization", computational linguistics, 28.4: 399-408, 2002
[11] Mani, I. "Automatic summarization", John Benjamins Publishing, - Computers - 285 pages 2001.
[12] H. P. Luhn. "The automatic creation of literature abstracts"IBM Journal of Research and Devlopment, 2(2):159–165, 1958.
[13] H.P. Edmundson. "New methods in automatic abstracting", Journal of the ACM (JACM) 16.2: 264-285, 1969.
[14] Jones, Karen Spärck. "Automatic summarising: The state of theart.", Information Processing & Management 43.6 1449-1481, 2007.
[15] Dang, H. and Owczarzak, K. "Overview of the TAC Update Summarization Task", In Proceedings of the Text Analysis Conference, TAC, Gaithersburg, 2008.
[16] Harman, D. "Overview of the First Text Retrieval" Conference (TREC-1). In Special Publication 500-207, Online Publication, TREC 92, pages 1–20, Gaithersburg, Maryland 1992.
[17] Harman, D. "Document Understanding Conference (DUC), Workshop on Text Summarization", New Orleans, Louisiana USA. In Conjunction with the ACM SIGIR Conference 2001
[18] Mani, I., House, D., Klein, G., Hirschman, L., Firmin, T., and Sundheim, B. "The TIPSTER SUMMAC Text Summarization Evaluation", In Proceedings of the 9th conference on European chapter of the Association for Computational Linguistics, EACL '99, pages 77–85, Bergen, Norway 1999.
[19] Wan, X. and Yang, J. "Single Document Summarization with Document Expansion". In Proceedings of the 22nd national conference on Artificial intelligence, volume 1 of AAAI'07, pages 931–936, Vancouver, British Columbia, Canada, 2007.
[20] Over, Paul, Hoa Dang, and Donna Harman, "DUC in context." Information Processing & Management 43.6:1506-1520, 2007.
[21] Hong, Kai, Mitchell Marcus, and Ani Nenkova. "System Combination for Multi-Document Summarization." Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 107–117, Association for Computational Linguistics, 2015.
[22] Mani, Inderjeet, and Eric Bloedorn. "Summarizing similarities and differences among related documents." Information Retrieval 1.1-2 :35-67, 1999
[23] Gunes Erkan and Dragomir R. Radev "LexPageRank: Prestige in Multi-Document Text Summarization" In EMNLP, Barcelona, Spain, 2004.
[24] Udo Hahn and Donna Harman, editors. Proc. of the Document Understanding Conference (DUC-02), Philadelphia, 2002.
[25] H. T. Dang. Overview of DUC 2006. In Proc. of the Document Understanding Conference (DUC 2006), 2006.
[26] Yihong Gong and Xin Liu. "Generic text summarization using relevance measure and latent semantic analysis", Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2001.
[27] Regina Barzilay and Lillian Lee. "Catching the drift: Probabilistic content models, with applications to generation and summarization", In Proc. of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'04), pages 113–120, 2004.
[28] Goldstein, Jade, et al. "Summarizing text documents: sentence selection and evaluation metrics." Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999.
[29] Leonhard Hennig, "Content Modeling for Automatic Document Summarization", PhD thesis, Berlin University, 2011.
[30] Vishal Gupta and Gurpreet Singh Lehal, "A Survey of Text Summarization Extractive Techniques", Journal of Emerging Technologies in Web Intelligence, 2.3, pp. 258-268, August 2010.
[31] Perera, Paththamestrige, "Syntactic sentence compression for text summarization", Master's thesis, Diss. Concordia University, 2013.
[32] Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown. "Introduction to the special issue on summarization", Computational Linguistic, 28(4):399–408, 2002.

[33] Udo Hahn and Inderjeet Mani, “The challenges of automatic summarization. Computer”, 33(11):29–36, 2000
[34] Ganesan, K., Zhai, C., and Han, J. “Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions”, In Proceedings of the 23rd International Conference on Computational Linguistics, COLING ’10, pages 340–348, Beijing, China, 2010.
[35] McKeown, K., and Radev, D. R. “Generating summaries of multiple news articles”, Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, United States. 74-82, 1995.
[36] Ehud Reiter and Robert Dale, “Building natural language generation systems”, Cambridge University Press, New York, NY, USA, 2000.
[37] DUC. Proc. of the document understanding conferences 2001- 2007. http://duc.nist.gov, 2007
[38] Abdullah Bawakid and Mourad Oussalah “A Semantic Summarization System” University of Birmingham at TAC 2008, Proceedings of the First Text Analysis Conference. Gaithersburg, MD, 2008.
[39] R. Mihalcea and P. Tarau, “TextRank: Bringing Order into Texts,” Association for Computational Linguistics, 2004.
[40] Xiaodan Zhang, “Exploiting External/Domain Knowledge to Enhance Traditional Text Mining Using Graph-based Methods”, Diss. Drexel University, 2009.
[41] H. Chim and X. Deng “Efficient Phrase - Based Document Similarity for Clustering”, IEEE Transactions on knowledge and data engineering, 20.9, 1217-1229, 2008.
[42] Yulia Nikolaevna Ledeneva, “Automatic language-independent detection of multiword descriptions for text summarization”, Diss. Instituto Politécnico Nacional. Centro de Investigación en Computación, 2008.
[43] D. R.Radev, H. Jing and M. Budzikowska “Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies” Information Processing and Management, 40.6, 919–938 – Elsevier, 2004.
[44] Rada Mihalcea and Paul Tarau “A Language Independent Algorithm for Single and Multiple Document Summarization” In Proceedings of international joint conference on natural language processing, IEEE, 2005.
[45] Chao Shen and Tao Li “Multi-Document Summarization via the Minimum Dominating Set” In Proceedings of the 23rd International Conference on Computational Linguistics, pages 984–992. Association for Computational Linguistics, 2010.
[46] Akiko Aizawa “An information-theoretic perspective of tf–idf measures” Journal of Information Processing and Management, 39.1,45–65, 2003.
[47] Shady shehata, Fakhri Karray, and Mohamed S. Kamel “An Efficient Concept-Based Mining Model for Enhancing Text Clustering” IEEE Transaction on Knowledge and Data Engineering, 22.10, 1360-1371, 2010.
[48] Pomikalek J. and Rehurek, R., “The influence of preprocessing parameters on text categorization”. International Journal of Applied Science, Engineering and Technology 1: 430-434, 2007.
[49] Songmei Cai and Zhao Lu “An Improved Semantic Similarity Measure for Word Pairs”, International Conference on e-Education, e-Business, e-Management and e-Learning, 2010.
[50] J. Sankari and Dr. R. Manavalan “Document Retrieval using Hierarchical Agglomerative Clustering with Multi-view point Similarity Measure Based on Correlation: Performance Analysis” International Journal of Scientific Engineering and Technology 2.9: 861-865, 2013.
[51] Khaled M. Hammouda and Mohamed S. Kamel “Incremental Document Clustering Using Cluster Similarity Histograms” Proceedings web intelligence, IEEE 2003.
[52] Ladda Suanmali, Naomie Salim and Mohammed Salem Binwahlan “Fuzzy Logic Based Method for Improving Text Summarization”, International Journal of Computer Science and Information Security 2.1, 2009.
[53] Mine Berker and Tunga Güngör “Using genetic algorithms with lexical chains for automatic text summarization” in Proc. of the 4th international conference on agents and aritificial intelligence, Vilamoura, Portugal, pp.595-600 2012.
[54] Shasha Xie and Yang Liu “Using corpus and knowledge-based measure in maximum marginal relevance for meeting summarization”, Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on. IEEE, 2008.
[55] Guangbing Yang, Kinshuk Kinshuk, Dunwei Wen and Erkki Sutinen “Enhancing Sentence Ordering by Hierarchical Topic Modeling for Multi-Document Summarization” Advances in Artificial Intelligence and Its Applications. Springer Berlin Heidelberg, 367-379, 2013.
[56] Chin-Yew Lin “Rouge: A package for automatic evaluation of summaries”, Text summarization branches out: Proceedings of the ACL-04 workshop. Vol. 8. 2004.
[57] Josef Steinberger, “Text Summarization within the LSA Framework”, PhD thesis, University of West Bohemia, 2007.
[58] Baeza-Yates and Ribeiro-Neto “Modern information retreival”, Vol. 463. New York: ACM press, 1999.
[59] Aminul Islam and Diana Inkpen, “Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity”, ACM Transactions on Knowledge Discovery from Data, 2.2, Article 10, 2008.
[60] Zhanying He, Chun Chen, Jiajun Bu, Can Wang and Lijun Zhang “Document Summarization Based on Data Reconstruction”, Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012.
[61] Yarlett, D. G., "Similarity-based generalization in language" P.81-83 Dissertation, Stanford University, 2008.
[62] Qiang LV and Ling SONG, “WordNet-based Methods for Short Context Similarity”, Journal of Pattern Recognition & Image Processing 4:3, 399-404, 2013.
[63] Ming Che Lee, JiaWei Chang and Tung Cheng Hsieh, “A Grammar-Based Semantic Similarity Algorithm for Natural Language Sentences”, Hindawi Publishing Corporation Scientific World Journal, Article ID 437162, 2014.
[64] Yihong Gong and Xin Liu “Generic text summarization using relevance measure and latent semantic analysis” Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2001.