Expert Systems with Applications 41 (2014) 5780–5787

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A multi-document summarization system based on statistics and linguistic treatment

Rafael Ferreira[a,b,*], Luciano de Souza Cabral[a], Frederico Freitas[a], Rafael Dueire Lins[a], Gabriel de França Silva[a], Steven J. Simske[c], Luciano Favaro[d]

[a] Informatics Center, Federal University of Pernambuco, Recife, Pernambuco, Brazil
[b] Department of Statistics and Informatics, Federal Rural University of Pernambuco, Recife, Pernambuco, Brazil
[c] Hewlett-Packard Labs., Fort Collins, CO 80528, USA
[d] Hewlett-Packard Brazil, Barueri, São Paulo, Brazil

A R T I C L E   I N F O

Keywords:
Multi-document summarization
Extractive summarization
Sentence clustering

A B S T R A C T

The massive quantity of data available today in the Internet has reached such a huge volume that it has become humanly unfeasible to efficiently sieve useful information from it. One solution to this problem is offered by using text summarization techniques. Text summarization, the process of automatically creating a shorter version of one or more text documents, is an important way of finding relevant information in large text libraries or in the Internet. This paper presents a multi-document summarization system that concisely extracts the main aspects of a set of documents, trying to avoid the typical problems of this type of summarization: information redundancy and diversity. Such a purpose is achieved through a new sentence clustering algorithm based on a graph model that makes use of statistic similarities and linguistic treatment. The DUC 2002 dataset was used to assess the performance of the proposed system, surpassing DUC competitors by a 50% margin of f-measure, in the best case.

1. Introduction

According to Kunder (2013), the estimated size of the web in 2013 was around 3.82 billion pages. This number grows every day at a fast pace, particularly regarding text documents (e.g. news articles, electronic books, scientific papers, blogs, etc.). Thus, it has become humanly unfeasible to efficiently sieve useful information from such a huge mass of documents. Automatic methods are needed to process the Internet data efficiently, scavenging useful information from it.

Text summarization (Wang, Li, Wang, & Deng, 2010) (TS) is a method that aims to create a compressed version of one or more documents, extracting the essential information from them. In other words, the goal of a summary is to present the main ideas in a document in less space (Radev, Hovy, & McKeown, 2002). A TS system is supposed to (i) identify the relevant contents from texts; (ii) eliminate redundant information; and (iii) keep a high level of coverage (He, Qin, & Liu, 2012) of contents.

TS can also be classified according to the number of documents simultaneously analyzed as *single* and *multi-document* summarization (Nenkova & McKeown, 2012). Multi-document summarization addresses the problems of text overload, as many documents share similar topics (Atkinson & Munoz, 2013).

In general, multi-document summarization is either *generic* (also termed *extractive*) (Alguliev, Aliguliyev, & Hajirahimova, 2012a) or *query-based* (Luo, Zhuang, He, & Shi, 2013). Generic summarization systems extract the main ideas from a text collection, while query-based ones select sentences related to a specific query performed by the user.

The same techniques used in single document summarization systems apply to multi-document ones; in multi-document summarization some issues as the degree of redundancy and information diversity increase, however. In a collection of texts on the same subject or a single topic (or a few topics), the probability of finding similar sentences is significantly higher than the degree of redundancy within a single text. Hence, anti-redundancy methods are crucial in multi-document summarization (Atkinson & Munoz, 2013). This issue is a well-known problem in TS: a good summary should avoid repeated information. Redundancy may

* Corresponding author at: Department of Statistics and Informatics, Federal Rural University of Pernambuco, Recife, Pernambuco, Brazil. Tel.: +55 8197885665.

E-mail addresses: rflm@cin.ufpe.br (R. Ferreira), lscabral@gmail.com (L. de Souza Cabral), fred@cin.ufpe.br (F. Freitas), rdl@cin.ufpe.br (R.D. Lins), gfps.cin@gmail.com (G. de França Silva), steven.simske@hp.com (S.J. Simske), luciano.favaro@hp.com (L. Favaro).

http://dx.doi.org/10.1016/j.eswa.2014.03.023
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

be perceived as a kind of "noise" that affects the quality of the final summary. On the other hand, summaries are supposed to encapsulate the maximum amount of information from texts (Goldstein, Mittal, Carbonell, & Kantrowitz, 2000) making possible the understanding of the main ideas from the original texts.

This paper proposes a new sentence clustering algorithm to deal with the redundancy and information diversity problems. The central assumption is that building a joint model of sentences and connections yields a better model to identify diversity among them (Cohn, Verma, & Pfleger, 2006). Based on that, the proposed algorithm uses the text representation proposed in Ferreira et al. (2013) to convert the text into a graph model containing four types of relations between sentences: (i) similarity statistics; (ii) semantic similarity; (iii) co-reference; and (iv) discourse relations. Such representation encapsulates the traditional approaches found in state of art systems (statistics similarity and semantic similarity) and linguistic treatment that improve the performance of redundancy elimination (Lloret & Palomar, 2013).

The proposed algorithm works as follows:

1. It converts the text into a graph model.
2. It identifies the main sentences from graph using Text Rank (Mihalcea & Tarau, 2004).
3. It groups the sentences based on the similarity between them.

The DUC 2002 conference dataset (NIST, 2002) was used to evaluate the algorithm presented, assessing it against the systems submitted to that conference. Two different experiments were conducted following the DUC 2002 guidelines: for each collection of documents, summaries with 200 words (first task) and 400 words (second task) were generated. The proposed system achieves Results 50% (first task) and 2% (second task) better than its competitors in terms of f-measure (Baeza-Yates & Ribeiro-Neto, 1999). It is important to stress that the DUC 2002 dataset is still the most widespread benchmark used today for multi-document summarization analysis and that to the best of the knowledge of the authors of this paper no other summarization system surpassed the performance figures published in NIST (2002).

The rest of this paper is organized as follows: Section 2 describes the main related work. Section 3 introduces the proposed system, its architecture and implementation. Section 4 presents an evaluation using the DUC 2002 conference dataset. Finally, some conclusions and discussion of lines for further work are presented in Section 5.

2. Related work

As already mentioned, multi-document summarization can be classified into *generic* and *query based* summarization. Currently, query based summarization has drawn a higher degree of interest in the community, due to its immediate applicability in commercial systems such as automatic customer services.

Traditional methods rely on statistics to create summaries. For instance, PRCN (Luo et al., 2013) is statistical framework to find *relevance*, *coverage* and *novelty* in multi-document summarization. It applies probabilistic latent semantic analysis (Hofmann, 1999) and probabilistic hyperlink-induced topic search (Cohn & Chang, 2000). PRCN attains good results regarding relevance (i.e., the identification of the key ideas of the text) and coverage (dealing with redundancy by excluding similar sentences). Canhasi and Kononenko (2014) models the query and the documents as a graph in order to increase the variability and diversity of the produced query-focused summary. It uses terms, sentences and documents as sets of vertices and the similarities among them as edges. The clusters are built based on the weight of the edges. Both works (Canhasi & Kononenko, 2014; Luo et al., 2013) are only suitable for query-based multi-document summarization.

Another example of this kind of summarizer is presented by Gupta and Siddiqui (2012). It combines single document summaries using sentence clustering techniques to generate multi-document summaries. It works as follows: (i) First, it creates a single document summary (using sentence scoring method); (ii) Then, it clusters the sentences using both syntactic and semantic similarities among sentences to represent the parts of the texts to be introduced in the summary; (iii) Finally, it generates the summary by extracting a single sentence from each cluster.

Canhasi and Kononenko (2014) proposed incorporating graphs to represent terms, sentences, documents and a query to improve coverage in query-based multi-document summarization. Goldstein et al. (2000) tries to minimize redundancy and to maximize both relevance and diversity. It first segments the documents into passages, and indexes them using inverted indices. After identifying the text passages which are relevant to the query using the cosine similarity, a number (depending on the compression rate) of sentences is selected. Finally, it reassembles the selected sentences into the final summary.

In the context of generic summarization, some systems must be highlighted. Radev, Jing, Stys, and Tam (2004) uses a cluster-based method to determine the relevance of sentences eliminating redundancy. His approach employs sentence scoring methods to select sentences for the summarization, achieving good results in redundancy detection. DESAMC + DocSum (Alguliev, Aliguliyev, & Isazade, 2012b) relies on genetic algorithms to create a summary taking into account several aspects of the text, namely *relevance*, *information coverage*, *diversity* and *length limit*. Atkinson and Munoz (2013) combines discourse-level knowledge and corpus-based semantic analysis to create summaries. Atkinson and Munoz (2013) claims that by employing rhetorical knowledge one obtains better quality summaries. The aforementioned approaches present valid contributions to the field and display good performance, in general. Chen, Jin, and Zhao (2014) uses a two-layer graph structure model to summarize documents. It uses the concept of a *phrase* as *related words* that appear together in a sentence. For example, "semitic western religion" and "Christian Philosophy" convey the same basic idea. The similarities among sentences are measured using cosine and co-occurrence of phrases similarities. The HITS algorithm (Wan, 2008) is used to identify the relevant sentences. All of these techniques use supervised learning, requiring a pre-annotated dataset, however.

Other works in generic summarization apply clustering methods to achieve larger information diversity, eliminating redundancy. Alguliev and his collaborators (Alguliev, Aliguliyev, & Mehdiyev, 2013) propose a generic document summarization method which is based on sentence-clustering. In their approach, sentences are represented as a *bag of words* and statistical and semantic similarities measure the dissimilarity among sentences. Such similarities are not combined. The authors use only one kind of similarity to perform the clustering process. A ranking-based sentence clustering framework is reported in Yang, Cai, Zhang, and Shi (2014). Differently of all previous work, it uses the information in documents, sentences and words to create clusters. In addition, that work proposes two different ranking functions (*simple* and *authority* ranking) to extract the main sentences from each cluster.

3. A new multi-document summarization algorithm

The multi-document summarization system proposed in this paper is based on statistical methods and linguistic treatment to increase information diversity of summaries also dealing with redundancy. The differences from the proposed system and the ones cited above are:

1. The system presented here generates a generic summarization and is completely unsupervised, while in the other systems (Alguliev et al., 2012b; Atkinson & Munoz, 2013; Radev et al., 2004) the user has to provide an annotated corpus.
2. The proposed system deals with redundancy and information diversity issues by clustering sentences (likewise Alguliev et al., 2013; Gupta & Siddiqui, 2012); nonetheless, the other systems rely only on syntactic and semantic similarities. The approach followed here also analyzes co-references and discourse relations.
3. The proposed system uses a graph model based on statistic similarities and linguistic treatment to represent the collection of input documents (differently from Canhasi & Kononenko, 2014; Chen et al., 2014).
4. The new graph model presented here minimizes co-references in most cases.
5. Statistic and semantic similarities, and linguistic operations such as co-reference resolution and discourse analysis are used here to perform summarization.

The system proposed here works as follows:

1. It opens all documents from an input collection and treat them as a single file.
2. It clusters sentences to find their relation to a specific topic.
3. It scores sentences to select the highest scored ones to compose each cluster.
4. It selects the main sentences from the clusters. The number of selected sentences depends upon the summarization rate provided by the user.

The following sections detail the sentence scoring methods and the sentence clustering algorithm used here. A sentence clustering example is presented together with explanations about the functioning of the proposed system. This system is readily amenable to additional machine learning since its many variables and be refined/updated as the system scales (particularly the number of ground truth documents scales).

3.1. Sentence scoring methods

This module is composed of two sentence scoring services. They were chosen because of their suitability in the summarization of texts of news articles (Ferreira et al., 2013). These services are detailed below:

Word frequency – This service carries out four steps: (i) It removes all stopwords; (ii) It counts the number of each word from the text (Word Frequency Score); (iii) For each sentence, it adds word frequency scores.

TF/IDF – This service encompasses the following steps: (i) Removing all stop words; (ii) Calculating TF/IDF for each word; (iii) For each sentence, it sums up the TF/IDF score of each word from it;

3.2. Sentence clustering

This section describes how sentence clustering is performed. It is divided into: the description of how the vertex and edges are created in the proposed approach; the details of the text clustering algorithm; and finally, a description of the configuration files necessary to create the sentence clusters.

3.2.1. Graph model

The proposed system relies on the *four dimension graph model* proposed in Ferreira et al. (2013).

The sentences from the document are seen as vertex and four different kinds of edges are used:

Statistical similarity: This method measures the content overlap between pairs of sentences. If it exceeds a threshold score, selected by the user, the edge between the sentence pair is created. Similarity is measured using the cosine metric. It creates a bag-of-words model to represent each sentence as an N-dimensional vector. The vector encapsulates the word and its frequency in the text. The similarity between two sentences is then defined by the cosine between the two corresponding vectors. Formula (1) calculates the cosine similarity.

$$Wsim(Si,Sj) = \frac{\vec{Si}.\vec{Sj}}{\left\|\vec{Si}\right\| x \left\|\vec{Sj}\right\|} \tag{1}$$

where, $\vec{Si}$ and $\vec{Sj}$ are weighted term-frequency vectors of sentence Si and Sj.

Semantic similarity: This measure addresses relations such as synonyms, hyponym, and hypernym (Varelas, Voutsakis, Raftopoulou, Petrakis, & Milios, 2005). For the semantic annotation of the sentences, four steps are accomplished:

1. Sentences are represented as a word vector, in which only nouns are retained;
2. The semantic similarity scores for each pair of words between two sentences is calculated;
3. The results are combined by summing up the scores;
4. The final scores are then normalized yielding a value between 0 and 1.

The measures of semantic similarity based on WordNet have been widely used in NLP applications (Pedersen, 2010). It is relevant to remark that semantic similarity is only calculated if instances of both words appear in WordNet, otherwise the score value for the pair is set to zero. In other words, this measure seeks the degree of similarity between two terms or words. These measures make use of the structure of WordNet to produce a numerical score that quantifies the degree of similarity between two concepts. In this work, the path metric is employed. This metric (Wubben & Bosch, 2009) computes the semantic relatedness of word senses by counting the number of nodes along the shortest path between the senses in the *is-a* hierarchies of WordNet. The path lengths include the end nodes. In other words, the longer the path length the less related the word is from the sense involved in the path. The relatedness value returned is the multiplicative inverse of the path length (distance) between the two concepts: relatedness = 1/distance. If two concepts are identical, then the distance between them is one; therefore, their relatedness is also 1.

Co-reference resolution: Co-reference resolution is the process by which different nouns that refer to a same entity are identified. The Stanford CoreNLP[1] tool was used here for indicating co-reference. When a co-reference is found, an edge in the graph with the relation is built.

Discourse relations: Relations between sentences and parts in a text are represented by *discourse relations*. The approach followed in this work was based on the results presented in Wolf and Gibson (2005), which presents a set of discourse structure relations based on contentful conjunctions. Some

[1] http://nlp.stanford.edu/software/corenlp.shtml.

examples are presented in Table 1, which were used to generate discourse relation edges.

3.2.2. Clustering algorithm

This module provides a six step algorithm to create text clusters. The input is a graph, usually obtained by using the previous module, and a configuration file (details in Section 3.2.3).

The pseudo code that follows (Fig. 1) presents an overview of the algorithm.

The explanation for each step is:

Input: The algorithm receives a graph and a configuration file as input. The graph describes the text by using the services proposed in the previous module. In other words, it uses sentences as vertices and creates the edges using a combination of the services provided above. The configuration file (details are found in Section 3.2.3) must contain the guidance information such as the threshold that measures the importance of a vertex.

TextRank score calculation: This step calculates the TextRank score for each vertex using the service provided by the summarization module. It extracts the keywords from a text document and also determines the weight of the "importance" of sentences within the entire document by using a graph-based model (Barrera & Verma, 2012; Mihalcea & Tarau, 2004).

Main vertex selection: It identifies the vertex with the highest TextRank scores.

Table 1
Discourse relations (Wolf & Gibson, 2005).

Cause-effect	because; and so
Violated expectation	although; but; while
Condition	if; as long as; while
Similarity	and; (and) similarly
Contrast	by contrast; but
Temporal sequence	then; before; after; while
Attribution	according to; claim that
Example	for example; for instance
Generalization	in general

```

input(Graph g, ConfigurationFile c)

For each vertex in g calculate the TextRank Score;

mainVertex = vertex with high TextRank Score;
LeadersVertices = {};

For each vertex in g different from mainVertex do
{
    if (c.threshold >= mainVertex.TextRankScore)
    {
        LeadersVertices = LeadersVertices + vertex;
    }
}

For each vertex in g different from LeadersVertices
  do
{
    Calculate the shortest path to all Leaders
        Vertex;
    closestLeader = Identifies the closest leader;
}

newGraphs = Remove all path to leaders different
    closestLeader;

return newGraphs;
```

Fig. 1. Clustering algorithm.

Leaders vertices selection: It uses the threshold value provided by the user and the TextRank scores to identify the leader vertices. Those vertices are used to create the clusters. In other words, each of the leader vertices creates one cluster. For example, if the main vertex has a TextRank score equal to 10 and the threshold is equal to 90%, all vertices with TextRank score higher than 9 will be selected as leader vertices.

Shortest path calculation: For each vertex, the algorithm calculates the shortest path between it and each leader vertices. It is calculated using Dijkstra's algorithm (Knuth, 1977).

Identifying the closest leader: In this step, each vertex identifies its closest leader vertex.

Removing the paths: It removes all paths that link a vertex to a leader, which are different from the closest leader identified in the previous step.

Output: As output, the system returns n graphs (where n is the number of leader vertices) representing n clusters.

3.2.3. Configuration

This module contains one class which encapsulates the text clustering parameters.

The graph construction application should consider the following:

Graph type: Selects if the graph is directed or undirected;
Edge selection: The edge creation services selected to the application;
Language selection: The language selected to the application;[2]
Domain selection: The domain selected to the application. Each domain particularities is treated.
Threshold: The threshold is a percentage (0%–100%) that will define which vertices are chosen as leaders.

3.3. Sentence clustering example

This section presents an example for a text clustering application. To simplify the example a cluster using only one text is created here. Fig. 2 shows the flow of activities in the system.

As detailed on Section 3.2.2 the flow of activities is divided into 6 steps. The input is a graph describing the text (sentences are vertices and edges are the relations presented in Section 3.2.1 and a configuration file that contains information like graph type, edge selection language, domain selection, and threshold). The output is a set of graphs with similar sentences.

The following sections detail the flow and provide an example for each step.

3.3.1. Input

The user provides the text file to be clustered and the configuration file as described in Section 3.2.3.

The configuration file is set as follows:

Graph type: Undirected;
Edge creation select: Similarity, Semantic similarity, and coreference;
Language selecting: English;
Domain Select: News;
Threshold: 90%.

A text extracted from CNN news page[3] was used in this case study. As already mentioned, the input to the text clustering

[2] This version of the framework provides semantic similarity, coreference resolution and discourse information only to texts in English.
[3] http://www.cnn.com/.

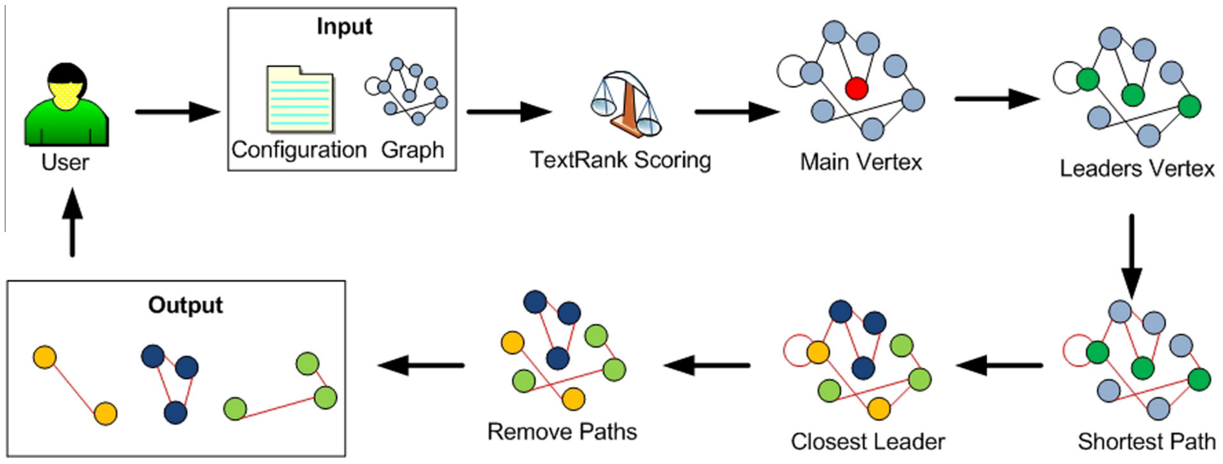

Fig. 2. Text clustering flow of activities.

application is a graph based on a text. Thus, a graph based on the following specifications is created:

- Each sentence is a vertex;
- Syntactical and semantic similarities, and co-reference resolution are used to create the edges.

Table 2 shows the text and a vertex number and Fig. 3 shows the graph representation.

3.3.2. TextRank scoring

This step calculates the TextRank score for each vertex. Table 3 shows the results.

3.3.3. Main vertex and leaders vertices

From Table 3, the main vertex selected is V2 and using the threshold from the configuration file (80%), one finds that V2 and V8 are the leader vertices.

3.3.4. Shortest path and closest leader

These steps find the cluster by choosing the closest leader for each vertex. For example, the path between V1 and V2 is shorter than V1 and V8.

3.3.5. Remove paths and output

The last step is removing the paths to the leaders different form the shortest one. The graphs obtained after the removal of the paths are the output. Figs. 4 and 5 show the output for the example presented.

4. Evaluation

There are several methods to evaluate a summarization output (Lloret & Palomar, 2012) in the literature. Relative utility (Radev & Tam, 2003), factoid score (Teufel, 2004), Pyramid method (Nenkova, Passonneau, & McKeown, 2007), are instances of some of them. However, the traditional information retrieval measures of *precision*, *recall* and *F-measure* are still the most popular evaluation method for such a task. Recall evaluates which number of the sentences selected by humans are also identified by a system, while precision is the fraction of those sentences identified by a system that are correct (Nenkova, 2006). F-measure is a combination of both precision and recall.

Lin (2004) proposed an evaluation system called ROUGE (Recall-Oriented Understudy for Gisting Evaluation). It provides a set of measures used for evaluating automatic summarization and machine translation using n-gram co-occurrence. The idea is to compare a summary or translation against a reference or a set of references, and count the number of n-grams of words they all have in common. Since 2004, ROUGE has become of widespread use for the automatic evaluation of summaries (Das & Martins, 2007; Wei, Li, Lu, & He, 2010).

The DUC dataset certainly constitutes one of the most representative dataset for summarization. The Document Understanding Conference (DUC)[4] carried out large-scale evaluations of summarization systems from 2001 to 2007. DUC provides different datasets for specific tasks (Lloret & Palomar, 2012). The 2002 conference was the last one that proposed the contest to create generic multi-document summaries. The dataset for multi-document summarization provided in 2002 consists of 567 documents divided into 59 collections (NIST, 2002). Two tasks related to extractive summarization were proposed in DUC 2002: the creation of summaries with 200 and 400 words.

The system proposed in this paper has reached a recall value of 62% and precision of 19% for the 200-word summary task, and a recall of 53% with precision of 17% for the 400- words one. These results show that the system presented here has a high coverage in 200 word summary (related to recall metric). Tables 4 and 5 display the results of the f-measure from the proposed system compared with the five best systems from the DUC 2002 conference.

The results of f-measure obtained show that the system proposed here surpasses DUC competitors by 50% in the first task (200 words) and by 2% in second task (400 words). Such encouraging results are probably due to the combination of statistics metrics with linguistic treatment. Co-reference resolution and discourse analysis certainly increase the quality of the generated summaries. The good recall stem from the good performance in removing redundancy by the clustering algorithm proposed.

5. Conclusions and lines for further work

This paper presents a multi-document summarization system created to deal with issues such as redundancy and information diversity. A new sentence clustering algorithm that identifies sentences related to different topics addressed in the documents to be

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

Table 2
Input text.

V0: A multi-million-dollar intrusion-detection system at New York's John F. Kennedy International Airport failed to notice a man who walked onto a runway Sunday,authorities said.
V1: The man, who was arrested after he was spotted by an airline employee, told police he was on a jet ski on Jamaica Bay adjacent to the runway and became stranded, according to Port Authority of New York & New Jersey official Anthony Hayes.
V2: The man climbed onto the tarmac from the water but the airport's security system did not detect him.
V3: The security system, called Perimeter Intrusion Detection System (PIDS), is manufactured by the Massachusetts-based Raytheon company.
V4: According to Raytheon's website, the mission of the $100 million system is to detect, assess and track intruders attempting to gain access into exterior secure areas".
V5: The system includes ground surveillance radars, video cameras with motion detection and smart fencing, according to Raytheon's website.
V6: In a statement, the Port Authority, which oversees the airport, called for "an expedited review of the incident and a complete investigation to determine how Raytheon's perimeter intrusion detection system, which exceeds federal requirements, could be improved.
V7: Our goal is to keep the region's airports safe and secure at all times.
V8: Raytheon would not comment on the incident.
V9: The airport is located in Jamaica Bay in the southeast section of Queens County in New York City.
V10: The runway where the man was spotted juts out into water that is surrounded by marsh and wetlands.
V11: The bay is easily accessible from the Rockaway Penninsula.
V12: According to a complaint filed by the Queens district attorney, a Delta Air Lines employee observed the man walking across the run way, he was arrested and charged with criminal trespassing.

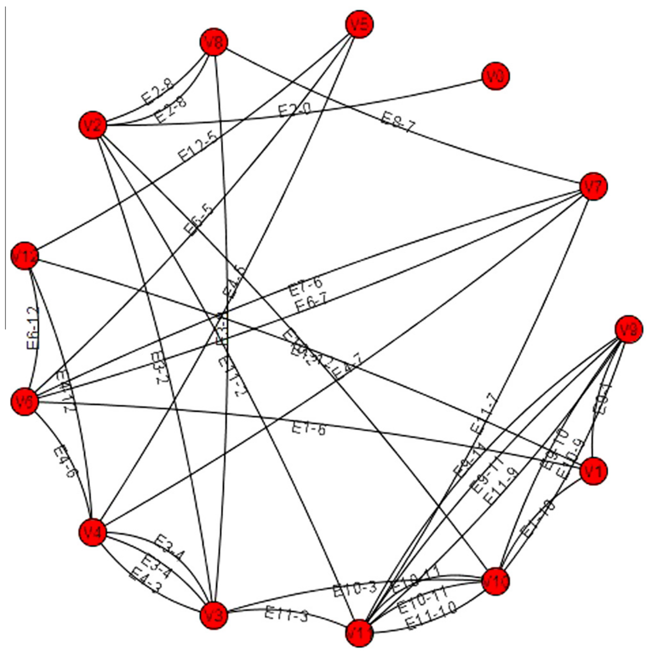

Fig. 3. Input graph.

Table 3
TextRank score for each vertex.

Vertex	TextRank score
V0	0.2435
V1	0.7054
V2	0.9821
V3	0.6743
V4	0.8643
V5	0.6669
V6	0.7196
V7	0.5584
V8	0.9401
V9	0.3912
V10	0.6198
V11	0.7952
V12	0.7710

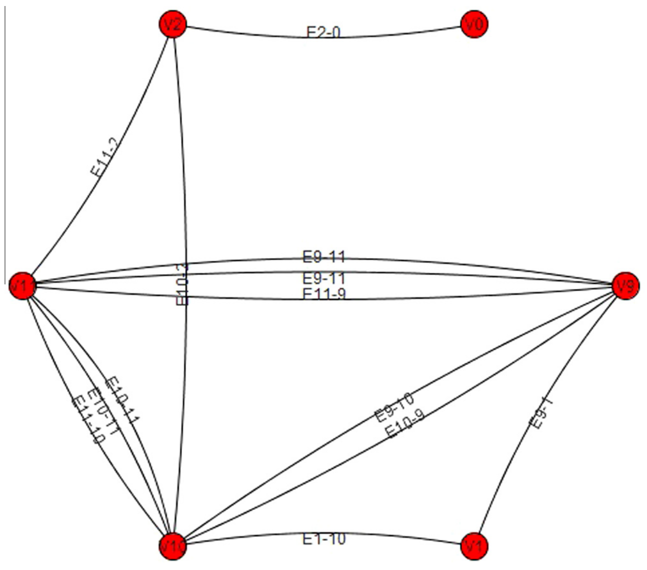

Fig. 4. Output 1.

summarized is proposed. A graph model is used to represent the documents using four different relations among sentences: (i) statistical similarity; (ii) semantic similarity; (iii) co-reference; and (iv) discourse relations.

The main contribution of the proposed algorithm is the creation of an unsupervised generic summarization system, that makes linguistic treatment of the input text by performing co-reference resolution and discourse analysis, besides using statistical and semantic similarities, as in all previous work in the literature. The new graph model employed by the system proposed detects co-references in most of the cases. The presented algorithm can be easily extended to perform clustering in different graph models.

Benchmarking the proposed algorithm using the DUC 2002 conference dataset showed that it outperformed all other DUC competitors. The F-measure of the proposed system was 50% better in the case of generating a 200-word summary from the DUC 2002 text collections.

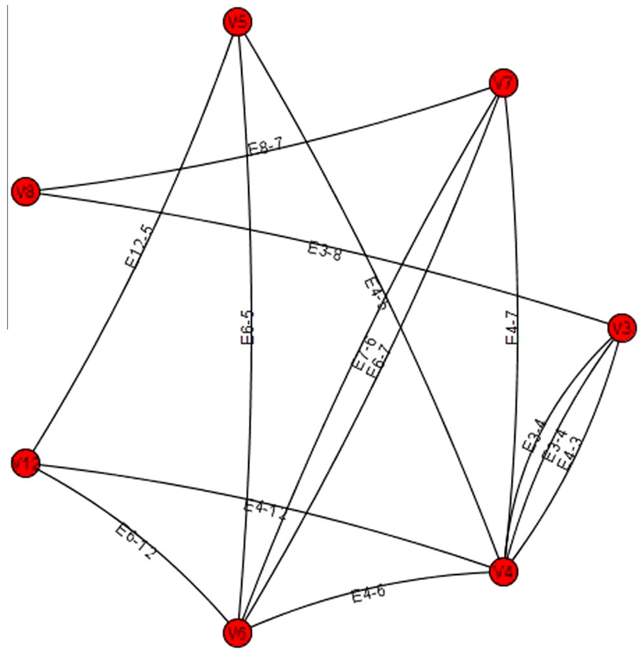

Fig. 5. Output 2.

Table 5
Comparison against DUC 2002 systems – 400 word summary.

System	f-Measure (%)
System 19	24
System 24	24,9
System 28	24,1
System 20	19,1
System 29	17,9
New System	25,4

Table 4
Comparison against DUC 2002 systems – 200 word summary.

System	f-Measure (%)
System 19	19,9
System 24	19,3
System 28	16,7
System 20	14,4
System 29	10,2
New System	30

The limitation of the approach proposed here is the problem of sentence ordering, as the system tries to find relevant sentences in groups of different topics; One possible way to overcome such a problem is to order the sentences using their relative position in the original documents and try to “align” the selected sentences. Besides that, the proposed system may be further improved in a number of ways, such as: benefitting from graph models to extract specific information from texts; and adding a language detection component to create a multilingual, multi-document TS system.

The application of the proposed system to increase the performance of information retrieval system is also being analyzed.

Acknowledgements

The research results reported in this paper have been partly funded by a R&D project between Hewlett–Packard do Brazil and UFPE originated from tax exemption (IPI – Law n 8.248, of 1991 and later updates).

References

Alguliev, R. M., Aliguliyev, R. M., & Hajirahimova, M. S. (2012a). Gendocsum+mclr: Generic document summarization based on maximum coverage and less redundancy. *Expert Systems with Applications, 39*(16), 12460–12473.
Alguliev, R. M., Aliguliyev, R. M., & Isazade, N. R. (2012b). Desamc+docsum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization. *Knowledge-Based Systems, 36*, 21–38.
Alguliev, R. M., Aliguliyev, R. M., & Mehdiyev, C. A. (2013). An optimization approach to automatic generic document summarization. *Computational Intelligence, 29*(1), 129–155.
Atkinson, J., & Munoz, R. (2013). Rhetorics-based multi-document summarization. *Expert Systems with Applications, 40*(11), 4346–4352.
Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval* (1st ed.). Addison Wesleys.
Barrera, A., & Verma, R. (2012). Combining syntax and semantics for automatic extractive single-document summarization. In *Proceedings of the 13th international conference on computational linguistics and intelligent text processing – volume part II, CICLing'12* (pp. 366–377). Berlin, Heidelberg: Springer-Verlag.
Canhasi, E., & Kononenko, I. (2014). Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization. *Expert Systems With Applications, 41*(2), 535–543.
Chen, H., Jin, H., & Zhao, F. (2014). Psg: A two-layer graph model for document summarization. *Frontiers of Computer Science, 8*(1), 119–130.
Cohn, D., & Chang, H. (2000). Learning to probabilistically identify authoritative documents. In *Proceedings of the seventeenth international conference on machine learning, ICML '00* (pp. 167–174). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
Cohn, D., Verma, D., & Pfleger, K. (2006). Recursive attribute factoring. In B. Schlkopf, J. Platt, & T. Hoffman (Eds.), *NIPS* (pp. 297–304). MIT Press.
Das, D., & Martins, A. F. T. (2007). A survey on automatic text summarization. Technical report. Literature survey for the language and statistics II course at Carnegie Mellon University.
Ferreira, R., Cabral, L. S., Lins, R. D., Silva, G., Freitas, F., Cavalcanti, G. D. C., et al. (2013). Assessing sentence scoring techniques for extractive text summarization. *Expert systems with applications, 40*(14), 5755–5764.
Ferreira, R., Freitas, F., Lima, R., Dueire, R., Cabral, L., Silva, G., et al. (2013). A four dimension graph model for automatic text summarization. In *Proceedings of IEEE/WIC/ACM international conference on web intelligence, WI13*. Atlanta, USA: ACM.
Goldstein, J., Mittal, V., Carbonell, J., & Kantrowitz, M. (2000). Multi-document summarization by sentence extraction. *Proceedings of the 2000 NAACL-ANLP workshop on automatic summarization, NAACL-ANLP-AutoSum '00* (Vol. 4, pp. 40–48). Stroudsburg, PA, USA: Association for Computational Linguistics.
Gupta, V. K., & Siddiqui, T. J. (2012). Multi-document summarization using sentence clustering. In *2012 Fourth international conference on intelligent human computer interaction (IHCI)* (pp. 1–5).
He, R., Qin, B., & Liu, T. (2012). A novel approach to update summarization using evolutionary manifold-ranking and spectral clustering. *Expert Systems with Applications, 39*(3), 2375–2384.
Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR '99* (pp. 50–57). New York, NY, USA: ACM.
Knuth, D. E. (1977). A generalization of Dijkstra's algorithm. *Information Processing Letters, 6*(1), 1–5.
Kunder, M. (2013). The size of the world wide web. Last Access February, (2014). <www.worldwidewebsize.com/?>.
Lin, C. Y. (2004). Rouge: A package for automatic evaluation of summaries. In M. Marie-Francine & S. Stan (Eds.), *Text summarization branches out: Proceedings of the ACL-04 workshop* (pp. 74–81). Barcelona, Spain: Association for Computational Linguistics.
Lloret, E., & Palomar, M. (2012). Text summarisation in progress: A literature review. *Artificial Intelligence Review, 37*(1), 1–41.
Lloret, E., & Palomar, M. (2013). Tackling redundancy in text summarization through different levels of language analysis. *Computer Standards & Interfaces, 35*(5), 507–518.
Luo, W., Zhuang, F., He, Q., & Shi, Z. (2013). Exploiting relevance, coverage, and novelty for query-focused multi-document summarization. *Knowledge-Based Systems, 46*(0), 33–42.
Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In *Conference on empirical methods in natural language processing*. Barcelona, Spain.
Nenkova, A. (2006). Summarization evaluation for text and speech: Issues Andapproaches. In *NTERSPEECH*.
Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. In *Mining text data* (pp. 43–76). Springer.
Nenkova, A., Passonneau, R., & McKeown, K. (2007). The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing, 4*(2), 1–23.
NIST, (2002). Document understanding conference. Last Access in September 2013. <http://www-nlpir.nist.gov/projects/duc/pubs.html>.

Pedersen, T. (2010). Information content measures of semantic similarity perform better without sense-tagged text. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics, HLT '10* (pp. 329–332). Stroudsburg, PA, USA: Association for Computational Linguistics.

Radev, D. R., Hovy, E., & McKeown, K. (2002). Introduction to the special issue on summarization. *Computational Linguistics, 28*(4), 399–408.

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

Radev, D. R., & Tam, D. (2003). Summarization evaluation using relative utility. In *Proceedings of the twelfth international conference on information and knowledge management, CIKM '03* (pp. 508–511). New York, NY, USA: ACM.

Teufel, S. (2004). Evaluating information content by factoid analysis: Human annotation and stability. In *EMNLP'04*.

Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E. G. M., & Milios, E. E. (2005). Semantic similarity methods in wordnet and their application to information retrieval on the web. In *Proceedings of the seventh annual ACM international workshop on web information and data management, WIDM '05* (pp. 10–16). New York, NY, USA: ACM.

Wan, X. (2008). Document-based hits model for multi-document summarization. In T. B. Ho & Z. H. Zhou (Eds.), *PRICAI. Lecture notes in computer science* (Vol. 5351, pp. 454–465). Springer.

Wang, S., Li, W., Wang, F., & Deng, H. (2010). A survey on automatic summarization. In *2010 International forum on information technology and applications (IFITA)* (Vol. 1, pp. 193–196).

Wei, F., Li, W., Lu, Q., & He, Y. (2010). A document-sensitive graph model for multi-document summarization. *Knowledge and Information Systems, 22*(2), 245–259.

Wolf, F., & Gibson, E. (2005). Representing discourse coherence: A corpus-based study. *Computational Linguistics, 31*(2), 249–288.

Wubben, S., & Bosch, A. (2009). A semantic relatedness metric based on free link structure. In *Proceedings of the eighth international conference on computational semantics, IWCS-8 '09* (pp. 355–358). Stroudsburg, PA, USA: Association for Computational Linguistics.

Yang, L., Cai, X., Zhang, Y., & Shi, P. (2014). Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. *Information Sciences, 260*, 37–50.