Expert Systems with Applications 40 (2013) 5755–5764

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Review

Assessing sentence scoring techniques for extractive text summarization

Rafael Ferreira [a,*], Luciano de Souza Cabral [a], Rafael Dueire Lins [a], Gabriel Pereira e Silva [a], Fred Freitas [a], George D.C. Cavalcanti [a], Rinaldo Lima [a], Steven J. Simske [b], Luciano Favaro [c]

[a] Informatics Center, Federal University of Pernambuco, Recife, Brazil
[b] Hewlett–Packard Labs., Fort Collins, CO 80528, USA
[c] Hewlett–Packard Brazil, Barueri, Brazil

ARTICLE INFO

Keywords:
Extractive summarization
Sentence scoring methods
Summarization evaluation

ABSTRACT

Text summarization is the process of automatically creating a shorter version of one or more text documents. It is an important way of finding relevant information in large text libraries or in the Internet. Essentially, text summarization techniques are classified as Extractive and Abstractive. Extractive techniques perform text summarization by selecting sentences of documents according to some criteria. Abstractive summaries attempt to improve the coherence among sentences by eliminating redundancies and clarifying the contest of sentences. In terms of extractive summarization, sentence scoring is the technique most used for extractive text summarization. This paper describes and performs a quantitative and qualitative assessment of 15 algorithms for sentence scoring available in the literature. Three different datasets (News, Blogs and Article contexts) were evaluated. In addition, directions to improve the sentence extraction results obtained are suggested.

1. Introduction

The rapid growth of the Internet yielded a massive increase of the amount of information available, especially regarding text documents (e.g. news articles, electronic books, scientific papers, blogs, etc.). Due to the huge volume of information in the Internet, it has become unfeasible to efficiently sieve useful information from the huge mass of documents. Thus, it is necessary to use automatic methods to "understand", index, classify and present all information in a clear and concise way, allowing users to save time and resources.

One solution is use text summarization techniques. Text summarization (TS) is the process of automatically creating a compressed version of one or more documents. It attempts to get the "meaning" of documents. Essentially, TS techniques are classified as *Extractive* and *Abstractive* (Lloret & Palomar, 2012). Extractive summaries produce a set of the most significant sentences from a document, exactly as they appear. Abstractive summaries attempt to improve the coherence among sentences by eliminating redundancies and clarifying the contest of sentences. It may even produce new sentences to the summary. Currently, the extractive summaries are commonly used because they are easier to create. Due to this in this work we focus on them.

Extractive methods are usually performed in three steps (Nenkova & McKeown, 2012):

- Create an intermediate representation of the original text;
- Sentence scoring;
- Select high scores sentences to the summary.

The first step creates a representation of the document. Usually, it divides the text into paragraphs, sentences, and tokens. Sometimes some preprocessing, such as stop word removal is also performed. The second step tries to determine which sentences are important to the document or to which extent it combines information about different topics, by sentence scoring. The score should be a measure of how significant a sentence is to the "understanding" of the text as a whole. The last step combines the score provided by the previous steps and generates a summary.

This paper describes 15 sentence scoring methods, and some variation of them, widely used and referenced in the literature applied to single document summarization in the last 10 years. Ani Nenkova points in Nenkova and McKeown (2012) three other types of sentence selection: bayesian topic models, sentence clustering, and domain-dependent topics. These methods are not explored in this paper, because the results yielded are not considered up to the same level as the others do not yet (Nenkova & McKeown, 2012). Each of the 15 scoring methods is described and implemented.

* Corresponding author. Tel.: +55 8197885665.
E-mail addresses: rflm@cin.ufpe.br (Rafael Ferreira), lscabral@gmail.com (L. de Souza Cabral), rdl@cin.ufpe.br (R.D. Lins), gfps.cin@gmail.com (G. Pereira e Silva), fred@cin.ufpe.br (F. Freitas), gdcc@cin.ufpe.br (G.D.C. Cavalcanti), rjlima01@gmail.com (R. Lima), steven.simske@hp.com (S.J. Simske), luciano.favaro@hp.com (L. Favaro).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.04.023

A quantitative and qualitative assessment of those methods using three different datasets (news, blogs, and articles context) is performed. The precision and recall measures (Baeza-Yates & Ribeiro-Neto, 1999) provided by ROUGE (Lin, 2004) were used to perform the quantitative assessment of the studied methods. The qualitative assessment was performed by four people who analyzed each original text and selected the sentences that they feel ought to be in the summary. The qualitative evaluation is done by counting the numbers of sentences selected by the system that match the human gold standard. Processing-time performance of each of the algorithm implemented is also taken into account.

It is important to notice that Lloret and Palomar (2012) and Nenkova and McKeown (2012) present two recent and comprehensive surveys on text summarization. They do not present any assessment of any sort of the techniques and this paper targets at filling in such an important gap.

In addition, some directions on "How Can Sentence Scoring Results be Improved?" are presented. Orasan (2009) and Nenkova and McKeown (2011) point that the main directions to do it are:

- Morphological transformation;
- There is often a large amount of words with little meaning to the text (stop words);
- The use of synonyms, words with similar semantics, may obscure the "weight" of a given word in the text in frequency-based methods;
- Co-reference;
- Ambiguity; and
- Redundancy.

This paper is structured as follows Section 2 presents the algorithms for sentence scoring more used in the technical literature. Section 3 explains the assessment parameters used. Section 4 presents the results of the quantitative, qualitative assessment of the algorithms together with the measures of time performance. Section 5 describes some problems that affect the results of the algorithms studied. In the conclusions, an account of the contribution made is presented together with lines for further work.

2. Sentence scoring methods

The first reference to text summarization using sentence scoring dates back to 1958 (Luhn, 1958; Lloret & Palomar, 2009). As already stated, the focus of these research areas are addressed by the following question: how can a system determine which sentences are representative of the content of a given text? In general, three approaches are followed: (i) *Word scoring* – assigning scores to the most important words; and (ii) *Sentence scoring* – verifying sentences features such as its position in the document, similarity to the title, etc.; and (iii) *Graph scoring* – analyzing the relationship between sentences.

The following section presents the main methods in each of the aforementioned approaches.

2.1. Word scoring

The initial methods in sentence scoring were based on words. Each word receives a score and the weight of each sentence is the sum of all scores of its constituent words. The approaches in the literature are outlined here.

2.1.1. Word frequency

As the name of the method suggests, the more frequently a words occurs in the text, the higher its score (Luhn, 1958; Lloret & Palomar, 2009; Gupta et al., 2011; Kulkarni & Prasad, 2010; Abuobieda, Salim, Albaham, Osman, & Kumar, 2012). In other words, sentences containing the most frequent words in a document stand a higher chance of being selected for the final summary. The assumption is that the higher the frequency of a word in the text, the more likely that it indicates the subject of the text.

2.1.2. TF/IDF

The hypothesis assumed by this approach is that if there are "more specific words" in a given sentence, then the sentence is relatively more important. The target words are usually nouns except for temporal or adverbial nouns (Satoshi et al., 2001; Murdock, 2006). This algorithm performs a comparison between the term frequency (*tf*) in a document (in this case each sentence is treated as a document) and the document frequency (*df*), which means the number of times that the word occurs along all documents. The *TF/IDF* score is calculated as follows:

$$TF/IDF(w) = DN\left(\frac{log(1+tf)}{log(df)}\right) \quad (1)$$

where *DN* is the number of documents.

2.1.3. Upper case

This method assigns higher scores to words that contain one or more upper case letters (Prasad et al., 2012). It can be a proper name, initials, highlighted words, among others. The score is calculated as follows:

$$CPTW(j) = \frac{NCW(j)}{NTW(j)} \quad (2)$$

where:

CPTW = Ratio of total first letter capital words present in the sentence to the total number of words present in the sentence,
NCW = Number of first letter capital words, and
NTW = Total number of words present in sentence.

$$UCf = \frac{CPTW(j)}{MAX(CPTW(j))} \quad (3)$$

where, *UCf* = Uppercase feature value.

2.1.4. Proper noun

Usually the sentences that contain a higher number of proper nouns are more important; thus, they are likely to be included in the document summary (Fattah & Ren, 2009). This is a specialization of the *Upper case* method.

2.1.5. Word co-occurrence

Word co-occurrence measures the chance of two terms from a text appear alongside each other in a certain order. One way to implement this measure is using *n*-gram (Mariòo et al., 2006), which is a contiguous sequence of n items from a given sequence of text or speech. In short, it gives higher scores to sentences that co-occurrence words appear more often (Liu, Webster, & Kit, 2009; Gupta et al., 2011; Tonelli & Pianta, 2011).

2.1.6. Lexical similarity

It is based on the assumption that important sentences are identified by strong chains (Gupta et al., 2011; Barrera & Verma, 2012; Murdock, 2006). In other words, it relates sentences that employ words with the same meaning (synonyms) or other semantic relation.

2.2. Sentence scoring

This approach analyzes the features of the sentence itself and was used for the first time in 1968 (Edmundson, 1969) analyzing the presence of cue words in sentences. The main approaches that follow this idea are described below.

2.2.1. Cue-phrases

In general, the sentences started by "in summary", "in conclusion", "our investigation", "the paper describes" and emphasizes such as "the best", "the most important", "according to the study", "significantly", "important", "in particular", "hardly", "impossible" as well as domain-specific bonus phrases terms can be good indicators of significant content of a text document (Gupta et al., 2011; Kulkarni & Prasad, 2010; Prasad et al., 2012). A higher score is assigned to sentences that contain cue words/phrases, using the formula:

$$CP = \frac{CPS}{CPD} \tag{4}$$

where,

CP = Cue-phrase score,
CPS = Number of cue-phrases in the sentence,
CPD = Total number of cue-phrases in the document.

2.2.2. Sentence inclusion of numerical data

Usually the sentence that contains numerical data is an important one and it is very likely to be included in the document summary, according to references (Fattah & Ren, 2009; Kulkarni & Prasad, 2010; Abuobieda et al., 2012; Prasad et al., 2012). This kind of sentence usually refers to some important information such as date of event, money transaction, damage percentage, etc.

2.2.3. Sentence length

This feature is employed to penalize sentences that are too short (Fattah & Ren, 2009) or too long (Abuobieda et al., 2012), these sentences are not considered as an optimal selection. The method uses length as number of words in sentence. In addition, Satoshi et al. (2001) penalizes sentences that are shorter than a certain length.

The first case could be calculated as follows:

$$Score = Length(s) * AverageSentenceLength \tag{5}$$

The penalty score is calculated using a conditional:

$$Score(Si) = \begin{cases} Li & \text{if } (Li > C) \\ Li - C & otherwise \end{cases} \tag{6}$$

where,

Li = length of sentence i and
C = certain length defined by user.

2.2.4. Sentence position

There are many approaches that use the sentence position as a score criterion (Fattah & Ren, 2009; Satoshi et al., 2001; Barrera & Verma, 2012; Abuobieda et al., 2012; Gupta et al., 2011). In reference (Abuobieda et al., 2012), the first sentence in the paragraph is considered an important sentence and a strong candidate to be included in the summary; Gupta et al. (2011) says that the first sentences of paragraphs and words in titles and headings are more relevant to summarization; The method proposed in reference (Satoshi et al., 2001) assigns score 1 to the first N sentences and 0 to the others, where N is a given threshold for the number of sentences.

Fattah and Ren (2009) follow the same principle as reference (Satoshi et al., 2001) and assume that the first sentences of a paragraph are the most important ones. The sentences are ranked as follows: the first sentence in a paragraph has a score value of 5/5, the second sentence has a score 4/5, and so on. Sentences further embedded in the paragraph are not significant. The latest approach in the literature (Barrera & Verma, 2012) exploits three position models. The first assumes that sentences closer to the start and end of a document are more likely to be more content representative. The second prioritizes only the top parts of the text. The last one uses sentences close to topic headings to create the summary.

2.2.5. Sentence centrality

Sentence centrality is the vocabulary overlap between a sentence and other sentences in the document (Fattah & Ren, 2009; Abuobieda et al., 2012; Kulkarni & Prasad, 2010). This approach makes no use any semantic treatment as Lexical Similarity. Another way to treat this measure is using other sentence similarity algorithms, for example, Bleu (Haque, Naskar, Way, Costa-jussa, & Banchs, 2010). Centrality could be calculated as follows:

$$Score = \frac{Ks \cap KOs}{Ks \cup KOs} \tag{7}$$

where,

Ks = Keywords in s, and
KOs = Keywords in other sentences.

2.2.6. Sentence resemblance to the title

Sentence resemblance to the title is the vocabulary overlap between this sentence and the document title (Satoshi et al., 2001; Fattah & Ren, 2009; Kulkarni & Prasad, 2010; Abuobieda et al., 2012). In this case, sentences similar to the title and sentences that include the words in the title are considered important. A simple way to calculate this score is:

$$Score = \frac{Ntw}{T} \tag{8}$$

where,

Ntw = Number of title words in sentence, and
T = Number of words in the title.

2.3. Graph scoring

In graph-based methods the score is generated by the relationship among the sentences. When a sentence refers to another it generates a link with an associated weight between them. The weights are used to generate the score of sentences.

2.3.1. Text rank

TextRank is a graph-based ranking model for text processing (Barrera & Verma, 2012; Mihalcea & Tarau, 2004). It extracts important keywords from a text document and also to determine the weight of the "importance" of words within the entire document by using a graph-based model. Sentences with a larger quantity of keywords get higher scores.

2.3.2. Bushy path of the node

The bushy path of a node (sentence) on a map is defined as the number of links connecting it to other nodes (sentences) on the map (Fattah & Ren, 2009).

2.3.3. Aggregate similarity

Aggregate similarity measures the importance of a sentence. Instead of counting the number of links connecting a node (sentence)

to other nodes (Bushy Path), aggregate similarity sums the weights (similarities) on the links (Fattah & Ren, 2009).

3. Evaluation parameters

This section describes: (i) the datasets used; (ii) methodology followed in the experiments to assess the quality of summaries; (iii) the computer used to perform the experiments.

3.1. Corpus

Three different datasets were used for testing the performance of the scoring methods presented. They are detailed in the following subsections.

3.1.1. CNN Dataset

The CNN corpus developed by Lins and colleagues (Lins et al., 2012) encompasses news articles from all over the world. The current version of this corpus presents 400 texts assigned to 11 categories: Africa, Asia, business, Europe, Latin America, Middle East, US, sports, tech, travel, and world news. The texts were selected from the news articles of CNN website (http://www.cnn.com). Besides the very high quality, conciseness, general interest, up-to-date subject, clarity, and linguistic correctness, one of the advantages of this new corpus is that a good-quality summary for each text, called the "highlights, is also provided. The highlights are three or four sentences long and are of paramount importance for evaluation purposes, as they may be taken as a summary of reference, or gold standard. In addition, two new summary evaluation sets were created. The first one was obtained by mapping the sentences in the highlights onto the original sentences of the text. The second one was generated by the authors blindly reading the texts and selecting n sentences that one thought better described each text. The value of n was chosen depending on the text size, but in general it was equal to the number of sentences in the highlight plus two. The most voted sentences were chosen and a very high sentence selection coincidence was observed. This second test set encompassed the first one in all cases.

3.1.2. Blog summarization dataset

In 2008, Hu and colleagues (Hu, Sun, & Lim, 2007, 2008) felt the need to get a Blog benchmark dataset. Thus, they decided to collect data from two blogs, Cosmic Variance (http://cosmicvariance.com) and Internet Explorer Blog (http://blogs.msdn.com/i.e./). Both have large numbers of posts and comments. From those blogs, 100 posts, 50 from each blog, were randomly chosen to form the evaluation dataset. To generate reference summaries, four human summarizers read the all chosen posts and their corresponding comments and then selected approximately seven sentences from each post.

3.1.3. SUMMAC dataset

The SUMMAC Corpus was elaborated under responsibility of the MITRE Corporation in cooperation with the University of Edinburgh, as part of the SUMMAC conference organizer group (Tipster Text Summarization Evaluation Conference) effort.[1] This dataset has 183 papers on Computation and Language, obtained from the repository LANL (Los Alamos National Laboratory) maintained by Cornell University Library, which currently holds more than 800,000 electronic documents from various fields in their database. After selecting the documents, they were annotated in XML, taking up their sections identified, being made available to Information Recovery, Extraction and Summarization Information can be obtained through the link: http://www-nlpir.nist.gov/related_projects/tipster_summac/cmplg-xml.tar.gz.

[1] http://www-nlpir.nist.gov/related_projects/tipster_summac/cmp_lg.html.

3.2. Evaluation methodology

This section describes the methodology followed in the experiments to assess the quality of summaries.

3.2.1. Quantitative assessment

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004) was used to quantitatively evaluate the summaries generated by using the different scoring methods. ROUGE is widely used for such purpose. This fully automated evaluator essentially measures the content similarity between system-developed summaries and the corresponding gold summaries.

The result of calculating ROUGE for the CNN dataset summaries is presented in two perspectives: (i) using the highlights of the CNN articles as the gold standards; and (ii) using the sentences that more closely match the highlights as the gold standards.

In relation to the blog summarization dataset all summaries (this dataset contains four summaries as presented in Section 3.1.2) are used as ROUGE input. At last, the SUMMAC Dataset provides the article abstract as input to ROUGE.

3.2.2. Qualitative assessment

The qualitative evaluation was performed in the CNN and Blog Summarization Dataset Corpora. As mentioned before, four people analyzed each original text and selected the sentences that they feel ought to be in the summary of those datasets. The qualitative evaluation is done by counting the numbers of sentences selected by the system that match the human gold standard. The SUMMAC Dataset provides only an abstract, which is not adequate to the assessment performed here.

3.3. Computer specification

To perform the experiment we use a computer with following specification:

- *Operational system*: Windows 7 64 Bits;
- *Processor*: Intel (R) Core (TM) i7-2670, 2.20 GHz;
- RAM memory: 8 GB.

4. Summarization performance evaluation

This section presents: (i) some abbreviations to better understand the experimentation; and (ii) details about the implementation of each method; and (iii) the results of the evaluation of the performance of the algorithms. The assessment was performed using each dataset separately.

4.1. Abbreviations

In order to facilitate presentation of the results, Table 1 lists the abbreviations to the terms used in next section and Table 2 shows a set of abbreviations for the name the algorithms.

4.2. Implementation of the algorithms

All algorithms described in Section 2 were implemented as described:

Word frequency: This algorithm is divided into four steps: (i) Remove all stop words; (ii) Count the number of each word from text. This step creates a structure that connects the word to the number of times it appears in the text (Word Frequency

Table 1
Abbreviations.

Average_R	Recall average
Average_P	Precision average
Average_F	F-measure average
Alg	Algorithm

Score); (iii) For each sentence it adds up the word frequency score of each word in a sentence.
TF/IDF: This algorithm is divided into the following steps: (i) Remove all stop words; (ii) Calculate the formula presented in Section 2.1.2 (TF/IDF Score) for each word from text; (iii) For each sentence it sum the TF/IDF score of each word in sentence.
TF/IDF: It is divided into: (i) Remove all stop words; (ii) Count the number of words with capital letters in text; (iii) Calculate the formula presented in Section 2.1.3 (Upper Case Score).
TF/IDF: The processing in this algorithm is performed as: (i) Remove all stop words; (ii) Perform POS tagging (using Stanford CoreNLP[2]) in order to select only nouns; (iii) Count the number of nouns that starts with capital letters (Proper Noun Score); (iv) For each sentence, add up the proper noun score of each word in a sentence.
Word co-occurrence: It is divided into: (i) Compute *n*-gram measure to n = 2, 3 and 4. (ii) For each sentence, add up the *n*-gram score of each word in a sentence.
Lexical similarity: It uses WordNet[3] to find similarity among words than applies Word Frequency algorithm.
Cue-phrases: There are three steps to perform this algorithm: (i) Load a cue-phrase list[4]; (ii) Count the total number of cue-phrases in the document; (iii) Calculate the formula presented in Section 2.2.1 (Cue-phrases Score) for each sentence from text.
Sentence inclusion of numerical data: This algorithm uses regular expressions to verify if some numerical data is present in sentences.
Sentence length: It works as follows: (i) Calculate the largest sentence length; (ii) Penalize sentences larger than 80 percent of the largest sentence length; (iii) Calculate the Sentence Length Score for all other sentences.
Sentence position 1 and 2: This algorithm combines the position score presented in Fattah and Ren (2009) and Barrera and Verma (2012). In short, the sentences are ranked as follows: the first sentence in a text has a score value of 5/5, the second sentence has a score 4/5, and so on. The same thing occurs with the last sentences: the last one receives score value of 5/5, penultimate has a score 4/5, and so on. The sentence position 1 applies this concept considering all text. On the other hand, sentence position applies to each paragraph from text.
Sentence centrality 1: It uses Bleu measure (from HultigLib[5]) in order to verify the similarity among sentences.
Sentence centrality 2: It implements the formula presented in Section 2.2.5.
Resemblance to the title: This algorithm implements the formula presented in Section 2.2.6.
Aggregate similarity: It follows two steps: (i) to create the link among sentences using the sum of all measures (from HultigLib); (ii) to sum all links score for each sentence.
TextRank Score: It uses the textrank algorithm provided in https://github.com/turian/textrank.

[2] http://nlp.stanford.edu/software/corenlp.shtml.
[3] http://wordnet.princeton.edu/.
[4] http://www.cs.otago.ac.nz/staffpriv/alik/papers/apps.pdf.
[5] http://www.di.ubi.pt/jpaulo/hultiglib/.

Table 2
Algorithms.

alg01	Word frequency
alg02	TF/IDF
alg03	Upper case
alg04	Proper noun
alg05	Word co-occurrence
alg06	Lexical similarity
alg07	Cue-phrase
alg08	Inclusion of numerical data
alg09	Sentence length
alg10	Sentence position 1
alg11	Sentence position 2
alg12	Sentence centrality 1
alg13	Sentence centrality 2
alg14	Resemblance to the title
alg15	Aggregate similarity
alg16	TextRank score
alg17	Bushy path

Bushy path: It is similar to aggregate similarity. Here, the algorithm counts the number of links differently from the previous one, which counts the link scores.

4.3. Assessment using CNN dataset

The result of calculating ROUGE for each algorithm, using CNN dataset as the gold standard, is shown in Table 3. Although the results obtained are close, some points should be remarked:

- Alg01, alg02 and alg09 achieved the best recall;
- Alg10, alg11, alg12 and alg14 reached the best precision;
- Alg01, alg02 and alg10 also achieved best *f*-measure;
- The Word scoring methods provided the best results of all the algorithms tested occupying the three of the top 5 positions in the assessment performed;
- The best word scoring algorithm was alg02;
- The best sentence scoring algorithm was alg10;
- The best graph scoring algorithm was alg16.

Fig. 1 presents the results of qualitative evaluation. As mentioned before, it counts the number of sentences selected by the system that match the human gold standard. The highest scores were obtained by: alg02 (611), alg01 (601), alg14 (580), alg06 (570), and alg09 (553).

Time performance of each algorithm is presented in Table 4.

Table 3
Results of ROUGE having CNN dataset as gold standard applied to the proposed algorithms.

	Average_R	Average_P	Average_F
alg01	0.71(0.19)	0.35(0.13)	0.46(0,15)
alg02	0.73(0.17)	0.35(0.12)	0.46(0.15)
alg03	0.64(0.19)	0.35(0.12)	0.44(0.12)
alg04	0.64(0.20)	0.35(0.13)	0.45(0.15)
alg05	0.59(0.20)	0.33(0.13)	0.42(0.15)
alg06	0.69(0.19)	0.35(0.13)	0.46(0.14)
alg07	0.50(0.22)	0.35(0.13)	0.40(0.14)
alg08	0.56(0.21)	0.36(0.13)	0.43(0.14)
alg09	0.70(0.18)	0.33(0.12)	0.44(0.15)
alg10	0.61(0.22)	0.40(0.13)	0.47(0.15)
alg11	0.52(0.22)	0.36(0.13)	0.41(0.12)
alg12	0.46(0.25)	0.37(0.16)	0.38(0.15)
alg13	0.33(0.21)	0.31(0.13)	0.30(0.15)
alg14	0.67(0.20)	0.36(0.12)	0.46(0.14)
alg15	0.57(0.20)	0.34(0.12)	0.42(0.14)
alg16	0.62(0.20)	0.34(0.12)	0.43(0.14)
alg17	0.56(0.20)	0.35(0.13)	0.42(0.14)

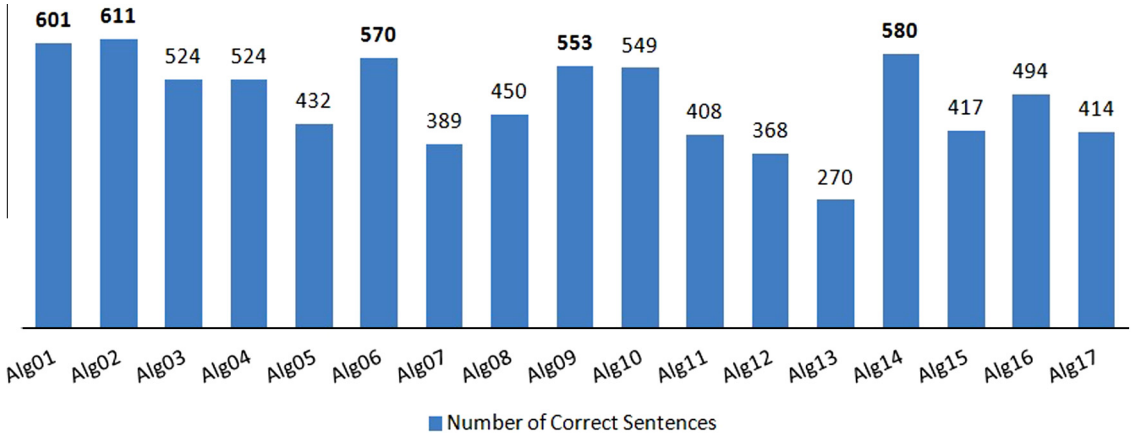

Fig. 1. Number of correct sentences *x* algorithms – using CNN dataset.

Table 4
Execution time using CNN dataset.

Alg	Execution time (s)
alg01	13.986
alg02	196.269
alg03	5.724
alg04	25.723
alg05	20.490
alg06	419.609
alg07	8.133
alg08	4.029
alg09	4.820
alg10	4.122
alg11	4.292
alg12	8.999
alg13	47.267
alg14	5.617
alg15	7.708
alg16	322.045
alg17	8.309

Table 5
Results of ROUGE having blog summarization dataset as the gold standard applied to the proposed algorithms.

	Average_R	Average_P	Average_F
alg01	0.72(0.13)	0.63(0.15)	0.67(0.14)
alg02	0.75(0.11)	0.63(0.15)	0.68(0.13)
alg03	0.58(0.16)	0.61(0.15)	0.59(0.15)
alg04	0.57(0.17)	0.63(0.14)	0.59(0.14)
alg05	0.65(0.14)	0.63(0.14)	0.63(0.13)
alg06	0.71(0.14)	0.63(0.14)	0.66(0.14)
alg07	0.52(0.18)	0.64(0.14)	0.57(0.15)
alg08	0.54(0.18)	0.63(0.15)	0.58(0.16)
alg09	0.76(0.11)	0.62(0.14)	0.68(0.13)
alg10	0.46(0.19)	0.60(0.13)	0.51(0.17)
alg11	0.52(0.18)	0.63(0.14)	0.56(0.16)
alg12	0.50(0.18)	0.65(0.14)	0.56(0.16)
alg13	0.46(0.20)	0.60(0.15)	0.51(0.18)
alg14	0.60(0.18)	0.64(0.13)	0.61(0.16)
alg15	0.58(0.18)	0.62(0.13)	0.59(0.16)
alg16	0.68(0.14)	0.63(0.14)	0.65(0.13)
alg17	0.58(0.17)	0.63(0.13)	0.60(0.15)

Some conclusions may be drawn from the results obtained:

- Combining the qualitative and quantitative assessments performed, one may conclude that the algorithms that yield better summarization are: alg01, alg02, alg14;
- The best result is obtained by alg02. It takes a longer time to execute than alg01 and alg14 (approximately 1,507 times longer than alg01 and 3,920 than alg14);
- Alg10 is the second fastest (behind only to alg08) and reaches a good quantitative results;
- Alg06 yields good results, but it is the slowest of all methods tested.

4.4. Assessment using the blog summarization dataset

The result of calculating ROUGE for each algorithm, using the blog summarization dataset, is shown in Table 5. Some points should be remarked:

- Alg01, alg02, alg06 and alg09 achieved the best results for recall;
- Alg12 and alg14 reached the best precision;
- Alg01, alg02 and alg09 also achieved the best f-measure;
- Again, the word scoring methods provided the best results of all the algorithms tested occupying three of the top 5 positions in the assessment performed;
- The best word scoring algorithm was alg02;
- The best sentence scoring algorithm was alg09;
- The best graph scoring algorithm was alg16.

Fig. 2 presents the results of the qualitative evaluation. As already explained, this assessment counts the number of sentences selected by the system that match the human gold standard. The highest scores were obtained by: alg09 (563), alg06 (552), alg16(551), alg01 (545), and alg02 (537).

Table 6 presents the time elapsed in the execution of the different scoring algorithms for the blog summarization dataset.

Conclusions:

- Combining qualitative and quantitative evaluations the best algorithms are: Alg02, Alg06, and Alg09;
- The best summarization results is provided by Alg09 and it is the fastest in relation to algorithms that provide the best summarization results.
- Alg06 archives good results, but it is the slowest amongst the algorithms tested.

4.5. Using SUMMAC dataset

The result of calculating ROUGE for each scoring algorithm, using as input the SUMMAC dataset with the gold standard, is shown in Table 7. Some points are worth remarking:

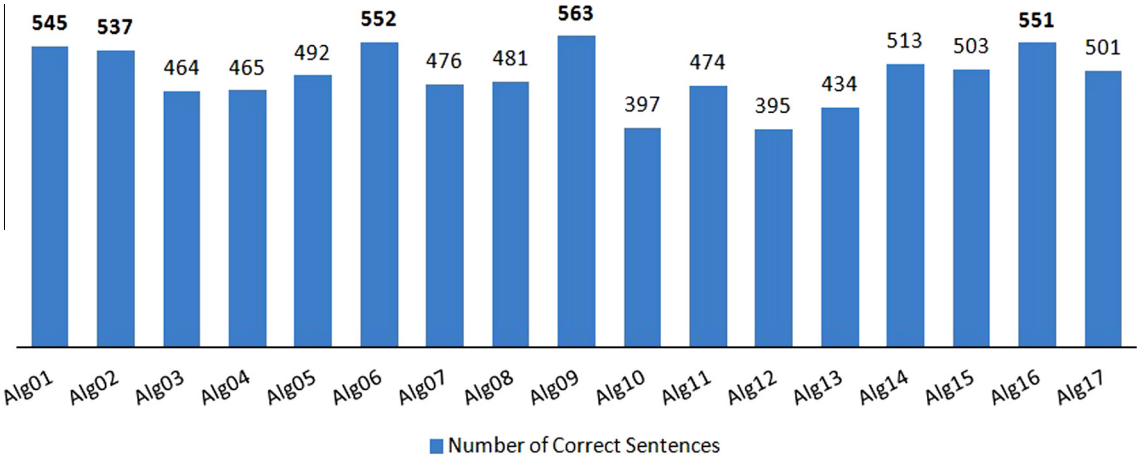

Fig. 2. Number of correct sentences *x* algorithms – using blog summarization dataset.

Table 6
Execution time using blog summarization dataset.

Alg	Execution time (s)
Alg01	2.508
Alg02	14.810
Alg03	1.841
Alg04	7.083
Alg05	2.943
Alg06	87.496
Alg07	2.982
Alg08	1.641
Alg09	2.391
Alg10	1.722
Alg11	1.799
Alg12	2.374
Alg13	5.225
Alg14	1.706
Alg15	2.194
Alg16	67.136
Alg17	2.508

Table 8
Execution time using SUMMAC dataset.

Alg	Execution time (s)
Alg01	17.287
Alg02	2,499.092
Alg03	9.606
Alg04	7.083
Alg05	73.948
Alg06	549.284
Alg07	29.954
Alg08	8.187
Alg09	9.670
Alg10	7.822
Alg11	7.750
Alg12	107.439
Alg13	2,602.518
Alg14	8.175
Alg15	38.316
Alg16	84.970
Alg17	39.903

Table 7
Results of ROUGE having SUMMAC dataset as gold standard applied to the proposed algorithms.

	Average_R	Average_P	Average_F
Alg01	0.48(0.10)	0.19(0.10)	0.26(0.10)
Alg02	0.47(0.11)	0.19(0.10)	0.26(0.10)
Alg03	0.25(0.11)	0.17(0.07)	0.19(0.06)
Alg04	0.22(0.11)	0.17(0.07)	0.18(0.06)
Alg05	0.23(0.16)	0.16(0.10)	0.17(0.10)
Alg06	0.46(0.11)	0.19(0.10)	0.26(0.10)
Alg07	0.33(0.11)	0.24(0.10)	0.26(0.07)
Alg08	0.25(0.11)	0.20(0.08)	0.21(0.07)
Alg09	0.49(0.09)	0.16(0.10)	0.23(0.09)
Alg10	0.31(0.10)	0.28(0.10)	0.28(0.06)
Alg11	0.24(0.11)	0.24(0.10)	0.22(0.08)
Alg12	0.07(0.11)	0.17(0.17)	0.07(0.08)
Alg13	0.22(0.11)	0.23(0.10)	0.21(0.08)
Alg14	0.36(0.14)	0.28(0.10)	0.29(0.08)
Alg15	0.22(0.08)	0.22(0.07)	0.21(0.05)
Alg16	0.46(0.10)	0.22(0.10)	0.28(0.09)
Alg17	0.23(0.10)	0.22(0.08)	0.21(0.06)

- Alg01, Alg02, and Alg09 achieved the best results for recall;
- Alg07, Alg08, Alg10, and Alg14 reached the best results for precision;
- Alg10, Alg14, and Alg16 also achieved the best *f*-measure;
- The sentence scoring methods provided the best results of all the algorithms tested occupying three of the top 5 positions in the assessment performed;
- The best word scoring algorithm was Alg02;
- The best sentence scoring algorithm was Alg14;
- The best graph scoring algorithm was Alg16.

To conclude the experiments Table 8 presents the results of the time execution evaluation on SUMMAC dataset.

Conclusions:

- The best results were obtained by algorithms: Alg10, Alg14, and Alg16;
- Alg16 is in the top-3 in summarization performance, but it is the fifth slowest one;
- Alg10 is the third fastest (behind to Alg04 and Alg11) and it also archived good quantitative results;
- Alg02 yields good summarization results, but it is the second slowest one.

4.6. Discussion

Faced on the results presented in this section we draw the following conclusions.

Considering the CNN dataset the results are reasonable because the documents are better structured. In summary:

- The documents use well-formed words, therefore the alg01 and alg02 archive good results;

- Generally, in news texts important phrases are at the beginning and end of document, and they are concise. It explains the good results of alg10, alg11 and alg09;
- The alg14 archive good results because the journalists usually provide titles containing the main information of the news;
- alg12 has good precision because these kind of texts tend to be slightly redundant.
- alg06 archive good qualitative result because it uses synonymous to choose the sentence.

Differently, from experiment using CNN dataset, in Blog Summarization Dataset, the sentence-based algorithms do not archive good performance, it is caused because in this type of text the writers are not concerned about the structure of text. Thus, algorithms like alg10 did not get result as good as in previous experiment.

The alg01, alg02, and alg06, once again, reached good results. It happens because, in general, social web tools (like blogs) are based in topic words. In summary, significant words in this kind of text are important.

Alg14 and alg09 have good results because blogs usually have small text, which implies: (i) that the title characterizes the text, and (ii) that the sentences are, in general, lower.

As happened all experiments, in the evaluation using SUMMAC dataset, the alg01 and alg02 archive good recall. The difference here is the alg09. It probably archive these recall because the authors usually use concise sentences to express main ideas.

The precision was higher in three sentence-based algorithms: (i) alg07, in scientific paper the authors use some cue words to contextualize the paper; (ii) alg10, the text, section and paragraph generally starts with the main sentence of the text; and (iii) alg14, The title need represent the text as good as possible.

The *f*-measure presents a surprise. Besides alg10 and alg14, the alg16 archive good results. It means that the relationship among sentences in the text are more important than in previous datasets.

Finally, in relation to execution time we find the following conclusions:

- Alg6 is the slower in almost all cases.
- The alg02 and alg13 greatly increased execution time in the last evaluation. This is mainly because they make computations with the words, and texts of the last dataset have more words.
- In general, sentence scoring methods are faster. This is because they use the sentence structure, unlike the word scoring (which makes computations using the words) and graph scoring (which creates a graph structure before running the algorithm).
- The blog dataset has the lower execution time because the texts here have fewer words in relation to the others.
- Alg16 usually is not fast because it has to create the graph and makes computations with words.

5. How can sentence scoring results be improved?

The sentence scoring algorithms are becoming increasingly mature. Consequently, the scientific community is now trying to improve their results rather than creating other algorithms. The six most common issues encountered are Orasan (2009), Nenkova and McKeown (2011): (i) Morphological transformation; (ii) Stop words; (iii) Similar semantics; (iv) Co-reference; (v) Ambiguity, and (vi) Redundancy. The following sections explain each of the strategies listed above and present possible solutions.

5.1. Morphological transformation

Constantin Orasan (2009) points three morphological transformations that improve word scoring methods.

Truncation: It retains only the first six characters of words are kept in an attempt to identify tokens derived from the same root.

Stemming: It is a transformation that builds the basic forms of words, i.e. strips off the plural "s" from nouns, the "ing" from verbs, or other affixes. A stem is a natural group of words with equal (or similar) meaning. After the stemming process, every word is represented by its stem. For instance, the verbs "traveling and "traveled are both transformed into "travel;

Lemmatization: This transformation identifies the lemma of a word. For example, it maps the verbs onto their infinitive and nouns onto their singular form. Thus, the form of the word must be known. Lemmatization requires more resources than the other two methods. It can deal with irregular words by using lists of exceptions.

Reference (Orasan, 2009) presents that the listed transformations improve the summarization results.

5.2. Stop words

The problem addressed here is how to deal with words with little meaning to the text, such as articles, conjunctions, and prepositions. Besides them, words with both high and low frequencies of occurrence are also considered as stop words. There are many tools, such as RetriBlog[6] and Lucene,[7] that provide the removal of stop words. It is important to notice that some stop words could be significant to text summarization, however. For example, some prepositions could refer to important text subjects (co-reference).

Almost all current summarization systems treat stop words (Lloret & Palomar, 2012; Barrera & Verma, 2012; Wei, 2012; Abuobieda et al., 2012) in some way.

5.3. Similar semantics

Words of similar semantics usually mean synonyms. However, relations such as *hypernyms* and *hyponyms* are also important to improve the semantic treatment. Hypernym relationships occur when words are related in some level of a semantic tree. For example, "pet and "dog. A "dog is a type of a"pet, thus they are related. In the problem of sentence scoring, words with similar semantics could be considered as one, increasing the relative importance that word as concept in the text.

There are three main approaches to deal with this problem. The first one is use WordNet[8] relations to verify the similarity between two given words. WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, each expressing a distinct concept. It also includes hypernyms and hyponyms. Reference (Pedersen, Patwardhan, & Michelizzi, 2004) provides an overview of similarity using WordNet. Some examples of summarization systems that use WordNet are (Lloret & Palomar, 2012; Barrera & Verma, 2012; Zhang, Ma, Niu, Gao, & Song, 2012; Gupta et al., 2011).

The second approach that deals with semantic similarity is known as *lexical chains* (Barzilay & Elhadad, 1997). This approach exploits the intuition that topics are expressed using not a single word but different related words, instead. For example, the occurrence of the words such as "car", "wheel", "seat", "passenger" indicates that the text is related to the automobile topic, even if each of the words does not appear very frequently in the text. In other words, this strategy clusters words together and the sentence scoring algorithms

[6] http://sourceforge.net/projects/retriblog/.
[7] http://lucene.apache.org/core/.
[8] http://wordnet.princeton.edu/.

analyze topics or concepts, rather than words in isolation. References (Wei, 2012; Gupta et al., 2011) use this approach.

The last approach is Latent Semantic Analysis (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). This is an unsupervised technique based on word co-occurrences to implicit represent text semantics, that tries to map which words usually appear together (Hachey, Murray, & Reitter, 2006; Balahur et al., 2009).

5.4. Co-reference

Co-reference is the process of matching all references to the same entity in a document, regardless of the syntactic form of the reference. It usually matches noun, full noun phrase or pronoun. Some work have demonstrated that co-reference resolution can be used for substantially improve summarization systems that rely on word frequency features (Nenkova & McKeown, 2011).

A simple example is the use of pronominal reference. For example, *"John will travel tomorrow. He bought the ticket yesterday"*. In this case the pronoun *"he"* refers to *"John"*. Thus, if the words are scored together, they may be more significant. This type of analysis is not widely used in summarization systems because of the performance and accuracy issues.

5.5. Ambiguity

Ambiguity, also known as polysemy, occurs when the same word can have different meanings in different contexts. For example,"apple" could mean a fruit or a computer company. Thus, the sentence score algorithms can assign higher values for some words improperly. Lexical chains may solve this problem.

Two fundamental issues must be taken into account in the context of summarization. Usually in single document summarization, words are in the same context. Here, the probability of ambiguity is low. On the other hand, in the context of multi-document summarization, such a problem may happen, but solving ambiguity may increase performance related problems.

5.6. Redundancy

Unlike the previously presented problems, redundancy is related to sentences and not only to words. Redundancy occurs when multiple sentences have the same content. In general, it is perceived as improper because of its use of duplicative or unnecessary wording, mainly in summaries.

The two techniques that are commonly used to treat this problem are:

Sentence fusion: It is the task of taking two sentences that contain some overlapping information, but that also have fragments that are different, and producing a sentence that conveys the information in common between the two sentences (Krahmer, Marsi, & van Pelt, 2008).

Textual entailment: It consists of determining if the meaning of one text snippet (the hypothesis) can be inferred by another one (the text) (Glickman, 2009). The identification of these entailment relations helps a summarization system avoid incorporating redundancy in final summaries.

These techniques are mainly used into abstractive summarization, but they may be adapted for extractive ones.

6. Conclusions

This paper explains and implements the most important text summarization strategies found in the literature in the last ten years. Three different corpora were used to assess the techniques presented. We selected the five best results obtained with the different test sets, one would obtain a coincidence of four methods as being the best ones: Word Frequency (Alg 1), TF/IDF (Alg 2), Lexical Similarity (Alg 6), and Sentence Length (Alg 9). The strategy "Text-Rank Score" (Alg 16) was also chosen by as providing good results for two of the three data sets tested. The results provided using ROUGE for the quantitative assessment of summarizers was quite close to the ones obtained by the qualitative analysis. The calculus of TF/IDF is by far the most computationally intensive of all methods tested (Alg 2). Methods Word Frequency (Alg 1) and Sentence Length (Alg 9) provide the best balance in execution-time performance and electing relevant sentences. Strategies to compose the results obtained to yield even better summaries are being currently investigated.

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

Abuobieda, A., Salim, N., Albaham, A. T., Osman, A. H., & Kumar, Y. J. (2012). Text summarization features selection method using pseudo genetic-based model. In *International conference on information retrieval knowledge management* (pp. 193–197).

Baeza-Yates, Ricardo, & Ribeiro-Neto, Berthier (1999). *Modern information retrieval* (1st ed.). Addison Wesley.

Balahur, Alexandra., Lloret, Elena., Boldrini, Ester., Montoyo, Andrés., Palomar, Manuel., & Martínez-Barco, Patricio. (2009). Summarizing threads in blogs using opinion polarity. In *Proceedings of the workshop on events in emerging text types* (pp. 23–31).

Barrera, Araly, & Verma, Rakesh (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* (pp. 366–377). Springer-Verlag.

Barzilay, Regina., & Elhadad, Michael. (1997). Using lexical chains for text summarization. In *Proceedings of the ACL workshop on intelligent scalable text summarization* (pp. 10–17).

Deerwester, Scott, Dumais, Susan T., Furnas, George W., Landauer, Thomas K., & Harshman, Richard (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

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

Fattah, Mohamed Abdel, & Ren, Fuji (2009). Ga, mr, ffnn, pnn and gmm based models for automatic text summarization. *Computer Speech and Language, 23*(1), 126–144.

Glickman, Oren (2009). *Applied textual entailment: A generic framework to capture shallow semantic inference*. VDM Verlag.

Gupta, P., Pendluri, V. S., & Vats. I. (2011). Summarizing text by ranking text units according to shallow linguistic features. In *13th International conference on advanced communication technology* (pp. 1620–1625).

Hachey, Ben., Murray, Gabriel., & Reitter, David. (2006). Dimensionality reduction aids term co-occurrence based multi-document summarization. In *Proceedings of the workshop on task-focused summarization and question answering* (pp. 1–7).

Haque, Rejwanul, Naskar, Sudip Kumar, Way, Andy, Costa-jussa, Marta R., & Banchs, Rafael E. (2010). Sentence similarity-based source context modelling in pbsmt. In *Proceedings of the 2010 international conference on asian language processing* (pp. 257–260). IEEE Computer Society.

Hu, Meishan., Sun, Aixin., & Lim, Ee-Peng. (2007). Comments-oriented blog summarization by sentence extraction. In *Proceedings of the 16th ACM conference on information and knowledge management* (pp. 901–904).

Hu, Meishan, Sun, Aixin, & Lim, Ee-Peng (2008). Comments-oriented document summarization: understanding documents with readers' feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 291–298). New York, NY, USA: ACM.

Krahmer, Emiel., Marsi, Erwin., & van Pelt, Paul. (2008). Query-based sentence fusion is better defined and leads to more preferred results than generic sentence fusion. In *Proceedings of the 46th annual meeting of the association for computational linguistics on human language technologies* (pp. 193–196).

Kulkarni, U. V., & Prasad, Rajesh S. (2010). Implementation and evaluation of evolutionary connectionist approaches to automated text summarization. In *Journal of Computer Science* (pp. 1366–1376). Science Publications.

Lin, Chin-Yew (2004). Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens (Ed.), *Text summarization branches out: Proceedings of the ACL-04 workshop* (pp. 74–81). Barcelona, Spain: Association for Computational Linguistics.

Lins, Rafael Dueire., Simske, Steven J., Cabral, Luciano de Souza., Silva, Gabriel de Frana., Lima, Rinaldo., Mello, Rafael F., & Favaro, Luciano. (2012). A multi-tool scheme for summarizing textual documents. In *Proceedings of 11st IADIS international conference WWW/INTERNET 2012* (pp. 1–8).

Liu, Xiaoyue, Webster, Jonathan J., & Kit, Chunyu (2009). An extractive text summarizer based on significant words. In *Proceedings of the 22nd international conference on computer processing of oriental languages. Language technology for the knowledge-based economy* (pp. 168–178). Berlin, Heidelberg: Springer-Verlag.

Lloret, Elena, & Palomar, Manuel (2009). A gradual combination of features for building automatic summarization systems. In *Proceedings of the 12th international conference on text. Speech and dialogue* (pp. 16–23). Berlin, Heidelberg: Springer-Verlag.

Lloret, Elena, & Palomar, Manuel (2012). Compendium: A text summarisation tool for generating summaries of multiple purposes, domains, and genres. *Natural Language Engineering*, 1–40 [FirstView].

Lloret, Elena, & Palomar, Manuel (2012). Text summarisation in progress: A literature review. *Artificial Intelligence Review, 37*(1), 1–41.

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

Mariòo, José B., Banchs, Rafael E., Crego, Josep M., Gispert, Adrià, Lambert, Patrik, Fonollosa, José A. R., et al. (2006). N-gram-based machine translation. *Computational Linguistics, 32*(4), 527–549.

Mihalcea, Rada., & Tarau, Paul. (2004). TextRank: Bringing order into texts. In *Conference on empirical methods in natural language processing, Barcelona, Spain*.

Murdock, Vanessa Graham. (2006). Aspects of sentence retrieval. Ph.D. thesis, University of Massachusetts, Amherst.

Nenkova, Ani, & McKeown, Kathleen (2011). Automatic summarization. *Foundations and Trends in Information Retrieval, 5*(2–3), 103–233.

Nenkova, Ani, & McKeown, Kathleen (2012). A survey of text summarization techniques. In *Mining text data* (pp. 43–76). Springer.

Orasan, Constantin (2009). Comparative evaluation of term-weighting methods for automatic summarization. *Journal of Quantitative Linguistics, 16*, 67–95.

Pedersen, Ted, Patwardhan, Siddharth, & Michelizzi, Jason (2004). Wordnet::similarity: Measuring the relatedness of concepts. In *Demonstration papers at HLT-NAACL 2004* (pp. 38–41). Stroudsburg, PA, USA: Association for Computational Linguistics.

Prasad, Rajesh Shardanand, Uplavikar, Nitish Milind, Wakhare, Sanket Shantilalsa, Jain, Vishal, Yedke & Tejas Avinash (2012). Feature based text summarization. *International Journal of Advances in Computing and Information Researches, 1*.

Satoshi, Chikashi Nobata., Satoshi, Sekine., Murata, Masaki., Uchimoto, Kiyotaka., Utiyama, Masao., & Isahara, Hitoshi. (2001). Keihanna human info-communication. Sentence extraction system assembling multiple evidence. In *Proceedings 2nd NTCIR workshop* (pp. 319–324).

Tonelli, Sara., & Pianta, Emanuele. (2011). Matching documents and summaries using key-concepts. In *Proceedings of the french text mining evaluation workshop*.

Wei, Yang. (2012). Document summarization method based on heterogeneous graph. In *9th International conference on fuzzy systems and knowledge discovery (FSKD)* (pp. 1285–1289).

Zhang, Dongmei., Ma, Jun., Niu, Xiaofei., Gao, Shuai., & Song, Ling. (2012). Multi-document summarization of product reviews. In *9th International conference on fuzzy systems and knowledge discovery (FSKD)* (pp. 1309–1314).