Expert Systems With Applications 65 (2016) 68–86

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization

Hilário Oliveira[a,*], Rafael Ferreira[a,b], Rinaldo Lima[a,b], Rafael Dueire Lins[a,b], Fred Freitas[a], Marcelo Riss[c], Steven J. Simske[d]

[a] Federal University of Pernambuco, Recife, Brazil
[b] Federal Rural University of Pernambuco, Recife, Brazil
[c] HP Brazil, Porto Alegre, Brazil
[d] HP Labs., Fort Collins, CO 80528, USA

ARTICLE INFO

Article history:
Received 22 April 2016
Revised 20 June 2016
Accepted 4 August 2016
Available online 9 August 2016

Keywords:
Automatic text summarization
Extractive-based approaches
Sentence scoring techniques
Single-document summarization
Multi-document summarization

ABSTRACT

The volume of text data has been growing exponentially in the last years, mainly due to the Internet. Automatic Text Summarization has emerged as an alternative to help users find relevant information in the content of one or more documents. This paper presents a comparative analysis of eighteen shallow sentence scoring techniques to compute the importance of a sentence in the context of extractive single- and multi-document summarization. Several experiments were made to assess the performance of such techniques individually and applying different combination strategies. The most traditional benchmark on the news domain demonstrates the feasibility of combining such techniques, in most cases outperforming the results obtained by isolated techniques. Combinations that perform competitively with the state-of-the-art systems were found.

1. Introduction

Alvin Toffler in 1970 coined the expression “information overload”, when he predicted that the exponential growth in the amount of information being produced would eventually cause people problems (Toffler, 1970). Such a scenario is the current reality. The Web, for example, allows creating, sharing, and accessing a vast amount of digital information, particularly textual documents such as news articles, online books, blogs, emails, scientific papers, tweets, among others. Despite the development of web search engines, sieving useful information from such massive volume of data is still a hard task, unfeasible to be performed manually. In such a context, there is a constant interest in tools capable of retrieving, classifying, and summarizing such information in an efficient manner.

In this scenario, Automatic Text Summarization (ATS) arises as a possible feasible solution to reduce users' time in identifying the most relevant information from a single document or a collection of text documents. ATS can be defined as the process of creating automatically a condensed version (summary) from a single- or multi-documents, while keeping their key information (Nenkova & McKeown, 2012). According to the Cambridge dictionary[1], a *summary* can be defined as “a short description that gives the main facts or ideas about something”. Based on this definition, an ATS system should deal with two fundamental issues (Saggion & Poibeau, 2013) **(i)** How to select the most relevant information, and **(ii)** Expressing the selected information in a compact way.

In general, ATS approaches have been classified in two major subfields *extractive* and *abstractive* (Lloret & Palomar, 2012). Extractive methods select the most relevant sentences in a document and use them to create the summary. Extractive summaries, due to selecting sentences *verbatim* from the original text, often present problems such as lack of coherence (Christensen, Soderland, Bansal, & Mausam, 2014), e.g., broken coreferences. On the other hand, *abstractive* approaches (Banerjee, Mitra, & Sugiyama, 2015; Khan, Salim, & Kumar, 2015) focus on selecting the most salient information fragments of a document and expressing them in a new form using operations such as sentence compression (Zajic, Dorr, Lin, & Schwartz, 2007) and merging (Filippova, 2010). Abstractive methods require a deep Natural Language Processing (NLP) analysis such as semantic representation and natural language generation. There are different methods to deal with abstractive summariza-

* Corresponding author.
E-mail addresses: htao@cin.ufpe.br (H. Oliveira), rflm@cin.ufpe.br (R. Ferreira), rjl4@cin.ufpe.br (R. Lima), rdl@cin.ufpe.br (R.D. Lins), fred@cin.ufpe.br (F. Freitas), marcelo.riss@hp.com (M. Riss), steven.simske@hp.com (S.J. Simske).

[1] http://dictionary.cambridge.org/us/

http://dx.doi.org/10.1016/j.eswa.2016.08.030
0957-4174/© 2016 Elsevier Ltd. All rights reserved.

tion such as semantic graph based method (Khan et al., 2015; Liu, Flanigan, Thomson, Sadeh, & Smith, 2015), multimodal semantic model (Greenbacker, 2011), among others. However, those methods are usually not completely automatic, as they require resources previously built, and demand a high computational effort. Due to these facts, extractive methods are more widely investigated today.

This work focuses on investigating extractive-based methods. Usually, this kind of method is performed in three steps (Nenkova & McKeown, 2012): **(i)** creation of an intermediate representation; **(ii)** computation of sentence salience (importance) scoring; and **(iii)** summary generation. Text documents are in an unstructured form; thus, it is necessary to pre-process these documents and represent them in a structured fashion. The first step usually involves some NLP tasks such as dividing the text into paragraphs, sentences, tokens, stopword removal, stemming, among others. Strategies to represent the main topic discussed in the document are also performed. Such strategies may compute the frequency or co-occurrence of words, sentence lengths and location into the document, presence of cue phrases, among others. The second step tries to estimate which sentences are the most relevant, based on the representation previously created. For each sentence a score is created, as a measure of its relevance. Finally, in the third step, the top-ranked sentences are selected to create the final summary. One of the most challenging issues in this step is to avoid redundancy, i.e., sentences with overlapping information in the summary.

Several extractive summarization techniques have been proposed and evaluated to estimate the relevance of a sentence. The techniques range from simple heuristics such as sentence position, sentence similarity with the document title, and statistical-based methods such as word frequency and co-occurrence. More sophisticated approaches such as clustering-based methods (Wan & Yang, 2008), graph-based methods (Mihalcea & Tarau, 2004), combinatorial optimization-based methods such as Integer Linear Programming (ILP) (Gillick & Favre, 2009; Li, Liu, & Zhao, 2015), supervised machine-learning approaches (Fattah, 2014), hierarchical approaches (Christensen et al., 2014), methods based on information extraction (Binh Tran, 2013), event-based summarization (Glavaš & Šnajder, 2014; Marujo et al., 2015), and semantic analysis (Baralis, Cagliero, Jabeen, Fiori, & Shah, 2013) have also been investigated.

This paper aims to investigate the performance of several shallow sentence salience scoring techniques widely used and referenced in the literature in the context of single- and multi-document summarization on the news domain. Different strategies to combine the individual scores of the techniques seeking to outperform the results obtained are also analyzed. The focus is in shallow sentence scoring techniques, i.e., heuristics or methods that are simple to implement and do not require massive computational effort to be computed. Experiments used the CNN corpus and the traditional DUC 2001–2004 datasets on both single- and multi-document summarization tasks. The results demonstrate that the performance of the features investigated and the combinations identified in terms of the most commonly used ROUGE evaluation measures (Lin, 2004) are feasible to identify the main gist of the documents, achieving comparable results against the state-of-the-art summarizers.

The main contributions of this paper are:

- Investigating several shallow sentence scoring techniques and ensemble strategies considering single- and multi-document summarization tasks in the most used datasets on the news domain.
- Showing that combining shallow sentence scoring techniques leads to an improvement in the performance of the summarization tasks based on the traditional ROUGE scores, in both single- and multi-document summarization tasks.
- Identifying combinations that perform competitively against several state-of-the-art systems on various benchmark datasets.

The remaining of this paper is organized as follows. Section 2 briefly presents the related works that assess several sentence salience scoring techniques. Section 3 introduces the summarization process adopted and the sentence salience scoring methods investigated in this work. Section 4 presents the results of the performed experiments. Finally, Section 5 presents the conclusions and draws lines for further work.

2. Related work

This section focuses on presenting the works that conducted studies either to compare the performance of the different sentence scoring techniques or the strategies to combine them in the context of extractive document summarization. The reader interested in an overview of ATS techniques may refer to the recent surveys in the field (Gambhir & Gupta, 2016; Lloret & Palomar, 2012; Nenkova & McKeown, 2012; Saggion & Poibeau, 2013; Torres-Moreno, 2014).

Meena and Gopalani (2014) investigated seven linear combinations using nine different sentence scoring techniques: Term Frequency - Inverse Document Frequency (TF-IDF), word co-occurrence, sentence centrality, sentence location, named entities frequency, the presence of positive and negative keywords, Textrank, and proper nouns frequency. The experiments used only ten documents of the Document Understanding Conferences (DUC) 2002 corpus[2]. The authors compared the performance of the combinations using the traditional ROUGE toolkit (Lin, 2004), which is extensively used to evaluate ATS systems. In a later work, Meena, Deolia, and Gopalani (2015) also investigated all possible linear combinations of six sentence scoring techniques. The authors assessed each combination using ten documents of the DUC 2002 dataset.

Ferreira et al. (2013) conducted an extensive assessment of seventeen sentence salience scoring techniques such as word frequency, TF-IDF, sentence centrality, sentence position, among others. The authors investigated the performance of these techniques individually using three different corpora on news, blog, and scientific paper domains. The authors complemented that study in another paper (Ferreira et al., 2014) analyzing the performance of ten proposed linear combinations using the seventeen features previously investigated on the three cited corpora. In both studies, each scoring technique and the proposed combinations were compared using the ROUGE toolkit and by counting the overlap of the sentences chosen by the methods in the automatically generated summaries and their *gold standards*, the extractive summaries created by experts using a computer-assisted methodology.

Other works addressed the sentence salience extraction task as a classification problem. They investigated the performance of many sentence scoring techniques as input features to Machine Learning (ML) algorithms. In such an approach, the problem consists of creating a classification model that estimates if a sentence should be included in the summary or not. Neto, Freitas, and Kaestner (2002) assessed thirteen sentence scoring techniques such as sentence length, sentence position, similarity to the title, among others, as input features to two well-known ML classification algorithms C4.5 (Quinlan, 1992) and Naive Bayes (John & Langley, 1995). Leite and Rino (2008) investigated the performance of several features based on both linguistic and statistical information, and complex networks to ATS using different ML algorithms. Fattah (2014) investigated eight shallow

[2] http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html.

scoring techniques including sentence centrality, sentence position, TF-IDF, among others, as features to the Maximum Entropy, Naive Bayes, Support Vector Machine, and a proposed hybrid machine learning model for multi-document summarization. Those authors conducted experiments on the news domain using the DUC 2002 dataset. Silva et al. (2015) investigated the performance of twenty sentence scoring techniques such as aggregate similarity, bushy path, sentence position, sentence similarity with the title, among others, as input features to different ML algorithms on news domain in the single-document summarization task. Silva and his colleagues applied feature selection algorithms to achieve a better setting of the training data and compared the performance against other summarizers.

Evolutionary algorithms such as Particle Swarm Optimization (PSO) (Binwahlan, Salim, & Suanmali, 2009) and Differential Evolution (DE) (Abuobieda, Salim, Kumar, & Osman, 2013) have also been investigated. Binwahlan et al. (2009) analyzed the performance of five sentence scoring features such as sentence centrality, title similarity, the presence of keywords, words frequency, and similarity with the first sentence. A PSO algorithm was applied to discover the best feature weights to improve the sentence extraction process. An experiment was set to validate the proposed method using one hundred documents from the DUC 2002 corpus. Abuobieda et al. (2013) introduced a combination of sentence scoring techniques such as sentence position, title similarity, sentence length, the presence of numerical data, and word frequency, with cluster-based methods in the context of the single-document summary task. DE algorithms were used to optimize the sentence clustering process using as similarity measure the classical Jaccard-Coefficient similarity. The performed experiments use the DUC 2002 dataset comparing the proposed approach with other summarizers.

All the aforementioned works have similar goals to the one presented here: investigating the performance of different sentence salience scoring techniques individually and, if possible, to improve the performance through the combination of different methods. This work seeks to overcome a few gaps identified in previous works such as: **(i)** The experiments conducted used, in most of the cases, only one corpus for each analyzed domain; **(ii)** Some of the works do not take into consideration the complete corpus, which can bias the conclusions, for instance, leading to a good local solution; **(iii)** Excepting for Meena et al. (2015) that analyzed all possible linear combinations of six sentence scoring techniques, the aforementioned works only evaluated a small number of proposed combinations; **(iv)** The lack of a comparison among different strategies to combine the sentence scoring methods; and **(v)** To the best of the knowledge of the authors of this paper, no previous work has conducted an extensive assessment of several shallow sentence scoring techniques in the context of multi-document summarization.

3. The single- and multi-document summarization processes

The key issue in extractive-based approaches is how to measure the representativeness of a sentence concerning the main topics discussed. Many features to compute the importance of a sentence have been proposed and evaluated in the literature (Barrera & Verma, 2012; Ferreira et al., 2013; Meena et al., 2015; Nenkova & McKeown, 2012). Those features are based on statistical, linguistic, and semantic techniques, taking into consideration different representation levels (Ferreira et al., 2013) such as *Word level* - assigning scores to the most relevant words; *Sentence Level* - verifying sentence features such as its position in the document, similarity to the title, and *Graph Level* - exploring the relationship among sentences or between words using graph algorithms.

Based on the general architecture identified in extractive summarization approaches (Nenkova & McKeown, 2012), the summarization process adopted here is presented in three steps:

Preprocessing: The raw textual input document or cluster of documents is preprocessed using well established NLP tasks. The well-known Stanford Natural Language Processing Toolkit (CoreNLP) (Manning et al., 2014) was used to carry out the NLP tasks of tokenization, sentence splitting, Part-Of-Speech (POS) tagging, lemmatization, named entity recognition, noun and verbal phrases extraction, and others.

Sentence scoring: In this step, the shallow sentence salience scoring techniques or combinations are applied to generate a score to each sentence in the input document or cluster. The techniques used in this step are described in the next subsections.

Summary generation: The sentences top-ranked on the score created in the previous step are iteratively inserted in the summary. To avoid redundancy, a new sentence to be included should have a degree of cosine similarity of less than 0.5 with each of the sentences already in the summary. The value of the threshold was empirically defined.

In the rest of this section, the eighteen sentence salience scoring techniques analyzed in this paper are briefly presented. The techniques were selected because they are widely used in literature, easy to implement, and do not require a high computational effort.

3.1. Aggregate similarity

Aggregate similarity (Ferreira et al., 2013) is a graphing technique based on the centrality idea that relevant sentences have more information in common with other sentences in a document. This technique computes the importance of a sentence s_i, which is a vertex in a graph, by summing up the weights of the edges it has with other vertices (sentences). Highly connected vertices may represent central sentences that indicate the main topics discussed in a document. One form of creating an edge between two sentences is to measure the similarity between them, and if this similarity is higher than a threshold, an edge between the two vertices is created. Eq. (1) shows how the salience of a sentence s_i is computed using this measure.

$$AggSim(s_i) = \sum_{j=1, i \neq j}^{S} edge\_weight \tag{1}$$

where,

- S is the total of sentences in the document,
- $edge\_weight$ is the similarity between the sentences s_i and s_j.

3.2. Bushy path

Bushy path (Fattah & Ren, 2009; Ferreira et al., 2013) is another graph-based method used to measure the salience of a sentence based on the centrality idea. It is very similar to the Aggregate Similarity method, but instead of summing up the similarities between the sentences, Bushy Path computes the importance of a sentence by counting the number of edges a sentence s_i has. This method score is computed as shown in Eq. (2).

$$BushyPath(s_i) = \frac{\#number\_of\_edges\_connected\_to\_s_i}{\#max\_number\_of\_edges\_connected\_to\_a\_sentence} \tag{2}$$

3.3. Cue-phrases

The cue-phrases method (Edmundson, 1969; Ferreira et al., 2013) is one of the first methods used for computing the importance of a sentence in ATS. The heuristic here is that sentences that contain cue-phrases such as “In summary”, “In conclusion”, “This report”, “The most important”, and others, have a higher probability to be included in the summary. This method relies on a pre-defined dictionary of cue phrases that can be discovered by analyzing summaries created by humans. This technique is computed as shown in Eq. (3).

$$CuePhr(s_i) = \frac{number\_of\_cue\_phrases\_in\_s_i}{total\_of\_cue\_phrases\_in\_the\_document} \tag{3}$$

3.4. Lexical similarity

Lexical similarity (Ferreira et al., 2013) is based on the assumption that key sentences are formed by strong chains of words, i.e., words with some semantic relation such as synonymy, hyponymy, and others. This method computes the similarity among the words in a sentence by assigning a higher weight to sentences with strong chains. The WordNet (Miller, 1995) is usually used to identify the semantic relations between words. Eq. (4) shows how this measure is calculated.

$$LexSim(s_i) = \frac{\#number\_of\_strong\_chains\_in\_s_i}{\#max\_number\_of\_strong\_chains\_in\_a\_sentence} \tag{4}$$

3.5. Named entities

Named Entity Recognition (NER) is the task of identifying and classifying entities in a text and associating them with semantic categories. Named Entities (NE) usually refer to real world entities such as the names of people, places, organizations, dates, time, numbers, among others. Such entities are important because they may describe the relevant actors in a document. This method computes the relevance of a sentence by counting the number of entities a sentence has as shown in Eq. (5). The main idea of this algorithm is that sentences encompassing more entities are more important to be included into the summary.

$$NER(s_i) = \frac{\#number\_of\_entities\_in\_s_i}{\#max\_number\_of\_entities\_in\_a\_sentence} \tag{5}$$

3.6. Noun and verbal phrases

A Noun Phrase (NP) is a group of nouns and its modifiers. In a sentence, a noun phrase can play the role of a subject, an object, or a complement. A Verbal Phrase (VP) consists of the main verb, supporting verbs, its complements, objects, or other modifiers. For instance, in the sentence “John Snow is going to 2014 FIFA World Cup Brazil”, the fragments “John Snow” and “2014 FIFA World Cup Brazil” are extracted as noun phrases and the fragment “is going to 2014 FIFA World Cup Brazil” is extracted as a verbal phrase. This method assumes that both noun and verbal phrases can identify important actors and facts discussed in a text respectively. The score of a sentence according to this feature is computed as presented in Eq. (6).

$$NP\_VP(s_i) = \frac{\#number\_of\_noun\_verb\_phrases\_in\_s_i}{\#max\_number\_of\_noun\_verb\_phrases\_in\_a\_sentence} \tag{6}$$

3.7. Numerical data

According to Ferreira et al. (2013), sentences containing numerical data are probably good candidates to be included in a summary. Usually numerical data refers to important fragments of information such dates, percentages, references to money, among others. Eq. (7) calculates the salience of a sentence s_i using this feature.

$$NumData(s_i) = \frac{number\_of\_numerical\_data\_in\_s_i}{total\_of\_words\_in\_s_i} \tag{7}$$

3.8. Open relations

Open Information Extraction (OIE) (Etzioni, Fader, Christensen, Soderland, & Mausam, 2011; Mausam, Schmitz, Bart, Soderland, & Etzioni, 2012) consists of extracting binary relational tuples rel(arg1, arg2) from a text, without requiring any training data. For instance, given the sentence “*Maurice Levy is the head of the one of the world’s largest communication firms*” an OIE system would extract the tuple *is the head of(Maurice Levy, one of the world’s largest communication firms)*. The presence of many relations in a sentence can be a clue to indicate that this sentence describes important facts. Based on this assumption, this method assigns a higher score to sentences containing more relations. This score is computed as presented in Eq. (8).

$$OpenRel(s_i) = \frac{\#number\_of\_open\_relations\_in\_s_i}{\#max\_number\_of\_open\_relations\_in\_a\_sentence} \tag{8}$$

3.9. Proper noun

Proper nouns may refer to important entities such as persons, places, organizations, among others. Sentences that contain more proper nouns can be considered as an important sentence, and it is most likely to be included in a document summary (Ferreira et al., 2013). This feature is similar to the named entities method previously described, the difference is that a named entity may be formed by one or more proper nouns, and depending on the efficiency of the NER tool adopted, the proper nouns may not be classified as a named entity. This feature score is calculated as presented in Eq. (9).

$$ProNoun(s_i) = \frac{number\_of\_proper\_nouns\_in\_s_i}{total\_of\_words\_in\_s_i} \tag{9}$$

3.10. Sentence centrality

The sentence centrality is defined as the degree of word overlap between a sentence s_i and other sentences in a document (Fattah & Ren, 2009; Ferreira et al., 2013). This method is based on the hypothesis that central sentences best describes the main information of a document. The sentence centrality score is given by Eq. (10).

$$SentCen(s_i) = \frac{W_{s_i} \cap W_{s_o}}{W_{s_i} \cup W_{s_o}} \tag{10}$$

where,

- W_{s_i} is the set of word in s_i,
- W_{s_o} is the set of word in the others sentences.

3.11. Sentence length

Selecting very short sentences may not represent the main topics of a document (Fattah & Ren, 2009; Ferreira et al., 2013). Similarly, selecting a very long sentence may be a waste of space considering that a sentence may contain important information in a part and non-relevant information in another. First, sentences smaller or larger than a given threshold are eliminated. Then, the score of the remaining sentences is given as presented in Eq. (11).

$$SentLen(s_i) = \frac{\#number\_of\_words\_in\_s_i}{\#max\_number\_of\_words\_in\_a\_sentence} \tag{11}$$

3.12. Sentence position

The position of a sentence in a document is one of the most effective heuristics to select relevant sentences for ATS, particularly for news articles (Edmundson, 1969; Ouyang, Li, Lu, & Zhang, 2010). The sentence position heuristic is that the first sentences in a document constitute the most relevant and their importances decrease as the sentence goes further in from the start of the text. Other approaches proposed variances assigning high importance to sentences at the beginning and at the end of the document (Ferreira et al., 2013). For long documents such as scientific articles or books, the sentence position can be counted at each paragraph.

Ferreira et al. (2013) assign more importance to sentences at both the beginning and the end of a document as follows: the first sentence has a score of $\frac{N}{N}$, the second sentence received a score of $\frac{N-1}{N}$, and so on, where **N** is a given threshold for the number of sentences taken into consideration. The same idea is applied, but starting from the end of the document. Abuobieda, Salim, Albaham, Osman, and Kumar (2012), on the other hand, use a strategy that assign a higher score to sentences only at the beginning of the document. This feature score is computed as shown in Eq. (12).

$$SentPos(s_i) = 1 - \frac{i}{S} \tag{12}$$

where,

- i is the ith sentence in the document, with i starting by zero,
- S is the total of sentences in the document.

3.13. Sentence resemblance to the title

The title usually reflects the main topics discussed in a document, especially in news articles. The sentence resemblance to the title technique measures the similarity among the sentences of a document and its title. This scoring technique assumes that sentences with a higher degree of similarity with the title indicate the main topic discussed in a document (Edmundson, 1969; Ferreira et al., 2013). For texts without title, some works consider the first sentence of the document as the title. This feature is computed as presented in Eq. (13).

$$SentRST(s_i) = \frac{W_{s_i} \cap W_t}{|W_t|} \tag{13}$$

where,

- W_{s_i} is the set of word in s_i,
- W_t is the set of word in the title,
- $|W_t|$ is the total of words in the title.

3.14. Term frequency - Inverse sentence frequency (TF-ISF)

TF-ISF is based on the classical Information Retrieval method Term Frequency - Inverse Document Frequency (TF-IDF). This variant of the original TF-IDF is applied to text summarization at sentence level instead of document level as TF-IDF. In this method, the frequency of a term t is computed in the entire document rather than in a specific sentence, whereas Inverse Sentence frequency measures how much descriptive a word is, i.e., if a word is common or rare across all sentences. This method assumption is that if a word is frequent and it appears in a few sentences, then it has a high probability to be included into the summary. The TF-ISF of a word is computed as shown in Eq. (14) and the salience score of a sentence is calculated as presented in Eq. (15).

$$TF - ISF(t_i) = TF(t_i) \times \log\left(\frac{S}{S_{t_i}}\right) \tag{14}$$

$$TF - ISF(s_i) = \sum_{t_j \in T}^{S} TF - ISF(t_j) \tag{15}$$

where,

- TF returns the frequency of a term t_i in the document(s),
- S is the total of sentences in the document,
- T is the total of terms in s_i,
- S_{t_i} is the total of sentences in which t_i occurs.

3.15. TextRank

The TextRank extraction algorithm is a graph-based ranking algorithm (Barrera & Verma, 2012; Ferreira et al., 2013; Mihalcea & Tarau, 2004) used to extract important keywords and determines the weight of these keywords within the entire document using a graph model. This method assigns a higher score to sentences containing many relevant keywords. Barrera and Verma (2012) state that optimal results can be found using only nouns and adjectives. The score of a term or n-gram, which is a vertex in the graph, is computed as presented in Eq. (16). The salience of a sentence s_i based on the TextRank algorithm is computed as shown in Eq. (17).

$$TextRank(v_i) = (1-d) + d \times \sum_{v_j \in In(v_i)} \frac{w_{ji}}{\sum_{v_k \in Out(v_j)} w_{jk}} \times TextRank(v_j) \tag{16}$$

$$TextRank(s_i) = \sum_{j=1}^{S} TextRank(t_j) \tag{17}$$

where,

- d is a damping factor that usually is set to 0.85 (Brin & Page, 1998).
- $In(v_i)$ is the set of vertices that point to v_i.
- $Out(v_j)$ is the set of vertices that v_j points to.
- S is the total of sentences in the document,
- $TextRank(t_i)$ returns the TextRank weight of a term or n-gram $t_i \in s_i$.
- w_{ji} is the number of co-occurrences between v_i and v_j.

3.16. Upper case

Upper case words are often more relevant than other words, as they may refer to important words such as acronyms, names of people, places, organizations, among others (Ferreira et al., 2013). This feature assigns higher scores to sentences containing more upper case words. The salience of a sentence s_i based on this method is computed as shown in Eq. (18).

$$UpperCase(s_i) = \frac{number\_of\_upper\_case\_words\_in\_s_i}{total\_of\_words\_in\_s_i} \tag{18}$$

3.17. Word co-occurrence

Word co-occurrence (Ferreira et al., 2013) measures the co-occurrence of one or more word sequences from a sentence appearing in other sentences. It assigns higher score to sentences with higher co-occurrence values. This method can be computed using n-gram similarity, which is a contiguous sequence of n items from a given sequence of text. Eq. (19) shows how this method score is calculated.

$$WordCoo(s_i) = \sum_{j=1, i \neq j}^{S} simNgram(s_i, s_j) \tag{19}$$

where,

- S is total of sentences in the document,
- sim returns the similarity of n-grams between two sentences s_i and s_j,
- N is the size of the words sequence.

Table 1
Statistics on the CNN and DUC datasets.

Corpus	Clusters	Documents	Sentences	Words	Task
CNN	0	3,000	115,649	2,628,336	Single
DUC 2001	30	309[a]	11,026	269,990	Single and Multi
DUC 2002	59	576[b]	14,370	348,012	Single and Multi
DUC 2003	30	298	7,691	197,483	Multi
DUC 2004	50	500	13,135	336,073	Multi

[a] Only the 308 distinct documents were used in the single-document experiments.
[b] Only the 533 distinct documents were used in the single-document experiments.

3.18. Word frequency

Word Frequency (Ferreira et al., 2013) is one of the oldest techniques to measure the relevance of a sentence for text summarization, first applied by Luhn (1958). It is based on the premise that relevant sentences contain more frequent words. In other words, the higher the frequency of a word, the more important it is to indicate the main topic of a document. Of course, not all words should be taken into consideration and, very often, stopwords filtering and stemming algorithm are applied before computing the frequencies. Some works (Abuobieda et al., 2012) select the first N more frequent words (in descendant order) in a document as Thematic Words. The sentence importance based on word frequency is given as shown in Eq. (20).

$$WordFreq(s_i) = \sum_{j=1}^{N} freq(w_j) \tag{20}$$

where,

- *freq* returns the frequency of a word w_i,
- N is the total of words in s_i.

4. Experimental evaluation

This section presents a comparative analysis of the eighteen sentence salience scoring techniques presented in the previous section. To assess their performance, several experiments were conducted to address the following issues **(i)** Evaluating of the sentence scoring techniques individually (Section 4.3); **(ii)** Assessing four strategies that combine the techniques (Section 4.4); **(iii)** Analyzing the individual techniques as input features to machine learning algorithms (Section 4.5); and **(iv)** Comparing the performance of the top performing individual techniques, combination methods, and machine learning algorithms, with the state-of-the-art summarizers (Section 4.6).

Before presenting the experimental results, a brief description of the experimental configuration is given in the next section, and some implementation decisions are introduced in Section 4.2.

4.1. Experimental setup

All experiments were performed in the context of single- and multi-document summarization in the news domain. In the single-document task, the DUC 2001–2002, and the CNN corpus were used. The multi-document task used the DUC 2001–2004 datasets. Some basic statistics of these corpora are shown in Table 1. The DUC datasets are the most widely used evaluation corpora for text summarization, mainly in generic news summarization. The CNN corpus is a new version of the dataset presented in Lins et al. (2012) and briefly described as follows:

- **CNN Corpus**. The CNN corpus is a collection of news document for single-document summarization based on the news articles from the CNN website (http://www.cnn.com). The current version of this corpus consists of 3000 articles in English distributed into twelve subject categories, originally tagged by CNN: Business, Health, Justice, Living, Opinion, Politics, Showbiz, Sports, Technology, Travel, United Stated, and world news. One important aspect of this corpus is the presence of a good-quality abstractive summary for each document written by the original authors, called *highlights*. The highlights served as the basis for the development of a *gold standard* extractive summary, developed by a team of experts using a computer-assisted methodology. The gold standards for the whole 3000 texts in the CNN-corpus encompass 10,754 sentences, around 10% of the total number of sentences. The experiments reported here were performed using the gold-standards of the CNN-corpus.
All DUC documents and clusters contain human models available with approximately 100 words, thus such size-threshold was used for the generated summaries.

Two evaluation measures were adopted:

- Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004) is a fully automated and widely used toolkit for text summarization evaluation. ROUGE measures the similarity between a set of candidate summaries with a collection of summaries models. ROUGE-N provides different levels of N-gram co-occurrences between the candidate summaries and the summary models. The most widely used ROUGE measures are ROUGE-1 and ROUGE-2, which compute the number of overlapping unigrams and bigrams, respectively. As suggested by Hong et al. (2014), ROUGE-1 recall has the highest recall ability to identify a better summary in a pair. In addition, Lin (2004) showed that ROUGE-1 recall achieves a high correlation with human judgments. ROUGE-2 recall with stemming and not removing stopwords was also computed, this setting presented the best agreement with manual evaluations as suggested by Owczarzak, Conroy, Dang, and Nenkova (2012). ROUGE-1 and ROUGE-2 recall were executed in all performed experiments. ROUGE-1.5.5 was set with parameters: -n 4 -m -c 95 -r 1000 -f A -p 0.5 -t 0. Since the DUC models have a threshold based on words counting, on these datasets, the parameter *-l N* was used to truncate all candidate summaries to N words.
- Direct Matching (DM) (Ferreira et al., 2013) is computed by counting the intersection of sentences between the candidate summary and the reference summary models available. An important aspect of this measure is its ability to identify methods that have high accuracy in recognizing the best sentences to compose a summary. This measure is only possible to be computed when extractive models are available. Thus, it is calculated particularly for the CNN corpus.

The statistical significance test applied in the experiments reported follows three steps: **(i)** The Shapiro-Wilk (Shapiro & Wilk, 1965) test was performed to verify the normality of the ROUGE-1 and ROUGE-2 recall distributions, **(ii)** If the distribution under testing fits a normal distribution the Paired T-test (Gibbons & Chakraborti, 2011) was applied; otherwise the two-side Wilcoxon signed-rank test (Gibbons & Chakraborti, 2011) is used; and **(iii)**

The selected statistical test in the previous step is performed twice: first using a null hypothesis of ‘equal averages’ and if $p - value < 0.05$ (5% of significance level), the test is performed again, but now using the null hypothesis of ‘higher or equal averages’. This methodology is applied in all performed experiments to assure if there is a significant difference between the techniques evaluated. All statistical tests in the experiments are executed using the statistical toolkit R[3].

4.2. Sentence scoring techniques settings

Techniques such as aggregate similarity, bushy path, sentence centrality, textrank, and word frequency, require the definition of a threshold setting, or a sentence similarity algorithm. Such decisions have a direct impact on the performance of those techniques. An experiment using the first version of the CNN corpus presented in Lins et al. (2012), was made to tune the parameters to support the following implementation decisions.

- **Aggregate similarity** and **Bushy pathy**: Both algorithms are implemented using the cosine similarity algorithm and adopting a similarity threshold equals to 0.1.
- **Lexical similarity**: The Resnik semantic similarity (Resnik, 1995) was used to measure the similarity between two words. It was computed using the WordNet (http://wordnet.princeton.edu/).
- **Sentence centrality**: Three different versions of this technique were evaluated by using: **(i)** *Sentence Centrality* (SentCen) which compute the similarity between two sentences by counting the intersection of words between them; **(ii)** *Sentence Centrality Cosine* (SentCenCos) which uses the cosine similarity algorithm; and **(iii)** *Sentence Centrality BLEU* (SentCenBLEU) (Haque, Naskar, Way, Costa-jussa, & Banchs, 2010) which compute the BLEU (Bilingual Evaluation Understudy) score to measure the similarity between two sentences. The BLEU measure implementation from (http://www.di.ubi.pt/~jpaulo/hultiglib/) was used.
- **Sentence length**: Remove the sentences with less than 10 non-stopwords and sentences containing more than 50 non-stopwords before computing the sentence score. The remaining sentences receive the score based on Eq. (11).
- **Sentence position**: This technique is implemented according to the two strategies presented in Section 3.12. *Sentence Position Version 1* (SentPosV1) uses the strategy presented in Ferreira et al. (2013), to assign higher scores to sentences at the beginning and at the end of the documents. On the other hand, *Sentence Position Version 2* (SentPosV2) (Abuobieda et al., 2012) gives more importance only to sentences at the beginning of the document.
- **Open Relations**: The open relations extracted from the traditional open information extractor system Reverb (Fader, Soderland, & Etzioni, 2011) are used to compute the score.
- **TextRank**: As suggested by Barrera and Verma (2012), a 4-gram model taking into consideration only nouns and adjectives is used to calculate this score.
- **Word Frequency**: As mentioned in Section 3.18 this algorithm can process all non-stopword frequencies to compute the relevance of a sentence or to define the N most frequent words as thematic words. ROUGE-1 recall was used as an evaluation measure to find the best results using $N = 100$.

4.3. Individual sentence scoring techniques evaluation

This experiment evaluates each sentence scoring techniques presented in Section 3 individually in both single- and multi-document summarization tasks. It aims to address the following empirical question: *Which are the best sentence scoring techniques for each corpus?*

Single-document summarization

Table 2 shows the results of the sentence scoring techniques evaluation in terms of ROUGE-1 recall (R-1), ROUGE-2 recall (R-2), and Direct Matching (DM). No variants of the same technique were selected, e.g., if Sentence Position Version 1 and version 2 are into the top-10 performing techniques, only the technique with the highest R-1 between them is selected. The same principle is applied to the Sentence Centrality, Sentence Centrality BLEU, and Sentence Centrality Cosine. The obtained results range substantially from one technique to another, showing that a wide variety of summaries are created.

Using the CNN corpus, TF-ISF achieves the top performance based on R-1 and R-2. It shows a significant improvement over all other techniques on both ROUGE scores at 95% of confidence level. Regarding the DM, sentence resemblance to the title and sentence position version 2 present the top-2 best results, recognizing 26.91% and 26.74% of the 10,754 sentences presented into the reference models. There is a high correlation between the top-10 best techniques selected based on DM and the R-1 scores. This is a positive fact, due to the ability of the DM measure to identify techniques that are more accurate in the sentence scoring task.

In the experiments performed using the DUC 2001 dataset, sentence position version 2 presents the best results taking into consideration both R-1 and R-2. Based on R-1, sentence position version 2 statistically outperforms all other techniques, except for the sentence resemblance to the title. Regarding R-2, sentence position version 2 shows a significant improvement over the remaining methods. Using the DUC 2002 corpus, sentence position version 2 achieves better results than all other techniques based on R-1 and R-2, presenting a significant improvement over all other methods. The sentence position heuristic has been shown to have a high performance on DUC datasets. In fact, it was used as a baseline in the DUC competitions in the single-document task.

In all three corpora, sentence position version 2 presents a better performance than version 1. This fact corroborates that the first sentences in news articles are usually the most important ones, and therefore have a higher probability to be included in the summary. Sentence position version 1 that also gives higher importance to sentences at the end of the document may be more suitable for another kind of document; for example, scientific papers or books, which are larger.

Usually, the title of a news article provides an important indication of the main topic discussed. The titles of the CNN documents are well-written and very descriptive; thus, sentence resemblance to the title achieves good performance in the ROUGE scores and the top performance in the DM score. The titles of the documents in the DUC datasets are not as descriptive as in the CNN documents; even so, the method sentence resemblance to the title yields a good performance in those datasets.

Other techniques such as aggregate similarity, bushy path, sentence centrality, sentence centrality cosine, textrank, TF-ISF, and word frequency also present good performance in all corpora. The centrality measures demonstrate that sentences sharing more information with other sentences are good candidates to be included in the summaries. Sentences containing more relevant words are also suitable candidates to be in the summary, the techniques of TF-ISF, word frequency, and textrank achieve success in recognizing the relevant words.

[3] http://www.r-project.org/.

Table 2
Results (%) and standard deviation (in parentheses) of the individual sentence scoring techniques evaluation on the single-document summarization task. The Top-10 highest techniques performance on each corpus are highlighted in bold. The top performing technique is indicated by * and the group of techniques statistically similar, if exist, is indicated by a †.

Techniques	CNN			DUC 2001		DUC 2002	
	DM	R-1	R-2	R-1	R-2	R-1	R-2
Aggregate Similarity	16.53	36.82 (17.64)	20.98 (19.29)	**41.16** (9.54)	**15.27** (10.10)	**43.73** (9.54)	**18.26** (10.08)
Bushy Path	**19.04**	41.11 (18.23)	24.28 (20.74)	**42.28** (9.69)	**16.31** (10.82)	**44.58** (9.25)	**18.90** (10.02)
Cue-Phrases	15.41	41.28 (18.58)	22.33 (21.26)	38.50 (9.40)	13.79 (9.60)	40.35 (9.67)	15.65 (9.10)
Lexical Similarity	16.92	45.62 (18.82)	25.65 (22.85)	38.59 (9.97)	13.59 (10.01)	41.27 (10.20)	16.09 (10.32)
Named Entities	**22.24**	**48.92** (19.88)	**31.33** (23.98)	**39.42** (10.40)	**15.11** (10.17)	**42.93** (10.21)	**18.07** (10.17)
Noun and Verbal Phrases	18.78	**50.70** (19.59)	**30.05** (24.78)	38.68 (10.39)	13.60 (9.80)	41.47 (10.08)	16.38 (10.04)
Numerical Data	12.80	31.99 (18.11)	17.74 (18.74)	35.65 (11.96)	12.68 (10.02)	37.25 (10.84)	14.21 (9.04)
Open Relations	**17.76**	**46.42** (19.54)	**26.97** (23.73)	**39.12** (10.27)	**14.41** (10.78)	**43.01** (9.77)	**17.57** (10.39)
Proper Noun	13.11	31.90 (17.10)	17.38 (18.05)	37.21 (10.05)	13.45 (9.55)	40.20 (10.34)	15.95 (9.97)
Sent. Centrality	**21.31**	**53.34** (19.62)	**32.75** (24.87)	40.26 (10.27)	14.86 (10.67)	43.70 (9.86)	18.18 (10.42)
Sent. Centrality BLEU	6.59	22.18 (18.82)	13.58 (19.38)	36.28 (10.44)	11.88 (9.63)	39.42 (9.66)	14.50 (9.60)
Sent. Centrality Cosine	18.46	40.73 (18.49)	23.99 (20.91)	**41.75** (9.10)	**15.79** (9.84)	**44.25** (9.25)	**18.72** (9.99)
Sent. Length	**19.32**	**52.67** (19.87)	**31.69** (25.75)	38.31 (10.38)	13.43 (10.19)	41.51 (9.85)	16.27 (9.98)
Sent. Position Version 1	18.17	39.00 (19.20)	24.68 (21.73)	41.49 (10.41)	17.61 (11.05)	44.38 (9.08)	19.72 (9.67)
Sent. Position Version 2	**26.74**	**45.99** (21.77)	**33.49** (25.00)	**43.75*** (10.47)	**19.57*** (11.64)	**46.94*** (9.20)	**22.14*** (10.01)
Sent. Resemblance Title	**26.91***	**49.29** (20.50)	**34.51** (23.67)	**42.59**† (10.20)	**17.93** (10.62)	**44.31** (10.74)	**19.95** (10.24)
TextRank	**21.99**	**49.74** (20.03)	**31.59** (24.36)	**40.66** (9.38)	**15.09** (9.76)	**43.93** (9.77)	**18.66** (10.11)
TF-ISF	**23.99**	**54.25*** (19.87)	**35.65*** (25.44)	**40.73** (10.46)	**16.01** (10.94)	**44.00** (9.72)	**18.70** (10.06)
Upper Case	12.04	29.54 (16.92)	16.31 (17.43)	37.47 (10.17)	13.65 (9.78)	40.57 (10.58)	16.36 (10.15)
Word Co-occurrence	13.28	41.74 (20.27)	22.71 (22.85)	37.17 (9.33)	12.20 (8.97)	38.85 (9.57)	13.80 (9.26)
Word Frequency	**24.34**	**52.40** (19.80)	**34.33** (24.90)	**41.83** (10.37)	**16.54** (11.01)	**44.77** (9.42)	**19.59** (9.96)

Table 3
Results (%) and standard deviation (in parentheses) of the individual sentence scoring techniques evaluation on multi-document summarization task. The performance of the top-10 techniques of each corpus are marked in bold. The top performing technique is indicated by a * and the group of techniques statistically similar is indicated by a †.

Techniques	DUC 2001		DUC 2002		DUC 2003		DUC 2004	
	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
Aggregate Similarity	**29.21**† (6.99)	**5.33**† (3.36)	**32.58**† (5.03)	**6.71**† (3.23)	**37.59**† (8.37)	**9.55*** (5.27)	**35.87**† (5.64)	**8.30**† (3.41)
Bushy Path	**29.67**† (6.21)	**5.12**† (3.12)	**32.67**† (5.36)	**6.49**† (3.11)	**36.96**† (7.28)	**8.39**† (5.03)	**36.85*** (4.97)	**8.34*** (3.75)
Cue-Phrases	**25.70** (5.42)	**3.41** (2.95)	**29.01** (5.98)	**4.37** (2.81)	29.95 (5.83)	5.03 (3.44)	28.92 (5.04)	4.06 (2.14)
Lexical Similarity	24.39 (5.39)	2.64 (1.90)	26.91 (6.52)	3.80 (2.76)	30.76 (6.09)	5.03 (2.57)	29.02 (5.99)	4.66 (2.80)
Named Entities	23.90 (6.69)	**3.52** (3.33)	26.83 (7.29)	**4.17** (2.90)	**33.22** (7.24)	**6.46** (3.47)	**30.75** (6.96)	**5.53** (3.24)
Noun and Verbal Phrases	**24.67** (6.24)	2.79 (2.64)	26.68 (6.29)	3.35 (2.29)	31.12 (7.14)	5.13 (4.03)	28.16 (5.21)	3.60 (1.73)
Numerical Data	21.94 (7.53)	3.08 (2.16)	24.72 (5.53)	3.26 (2.30)	28.78 (7.75)	5.85 (4.32)	26.99 (6.30)	4.06 (3.06)
Open Relations	**25.78** (4.92)	2.99 (1.91)	**28.43** (5.26)	3.84 (2.77)	32.24 (6.41)	5.57 (3.46)	**31.10** (4.10)	4.88 (2.31)
Proper Noun	20.42 (6.52)	2.84 (2.54)	23.34 (6.76)	4.15 (2.77)	28.67 (7.21)	5.38 (3.60)	28.52 (5.60)	5.02 (2.79)
Sent. Centrality	26.34 (7.33)	3.62 (4.35)	28.73 (6.42)	4.32 (3.02)	33.75 (6.21)	6.63 (4.50)	30.22 (5.85)	4.69 (2.60)
Sent. Centrality BLEU	22.35 (5.55)	1.92 (1.14)	24.11 (6.31)	2.76 (2.05)	23.83 (6.10)	3.30 (2.42)	25.88 (5.14)	3.15 (1.85)
Sent. Centrality Cosine	**30.25**† (6.18)	**5.50**† (3.37)	**33.33**† (4.97)	**7.29*** (3.14)	**37.61*** (8.00)	**9.42**† (5.43)	**36.69**† (4.45)	**8.31**† (3.23)
Sent. Length	25.12 (6.00)	2.97 (2.77)	27.01 (6.61)	3.68 (2.79)	31.96 (5.10)	5.51 (3.90)	27.95 (5.29)	3.66 (2.26)
Sent. Position Version 2	**30.29**† (5.93)	**5.10**† (3.98)	**33.78*** (5.65)	**7.16**† (3.69)	**35.93**† (7.23)	**7.67** (3.98)	**35.78**† (4.52)	**7.87**† (3.17)
Sent. Resemblance Title	**30.63*** (6.66)	**5.94*** (3.74)	**32.60**† (6.17)	**6.87**† (3.56)	**35.79**† (6.87)	**7.84** (5.23)	**35.45**† (4.92)	**7.91**† (3.20)
TextRank	**28.72** (6.80)	**5.00** (3.40)	**33.62**† (6.09)	**6.98**† (3.95)	**35.75**† (7.08)	**6.89** (3.85)	**35.85**† (5.19)	**7.73**† (3.26)
TF-ISF	**27.79** (8.18)	**4.71** (4.09)	**30.08** (7.85)	**5.58** (3.68)	**35.64**† (7.29)	**7.69** (5.62)	**35.55**† (6.57)	**8.18**† (3.58)
Upper Case	21.11 (6.52)	3.02 (2.37)	23.32 (6.51)	3.95 (2.54)	29.01 (6.79)	5.73 (3.61)	28.48 (5.72)	**5.31** (2.85)
Word Co-occurrence	23.72 (5.66)	2.60 (1.88)	26.40 (6.59)	3.75 (3.04)	**33.18** (7.01)	**6.04** (4.36)	30.06 (5.85)	4.88 (2.69)
Word Frequency	**28.28** (7.15)	**4.97**† (3.95)	**30.17** (7.82)	**5.39** (3.28)	**36.68**† (7.66)	**8.41** (5.68)	**35.83**† (4.61)	**8.15**† (3.07)

Multi-document summarization

Table 3 presents the individual sentence scoring techniques performance on the multi-document summarization task. Since the method sentence position version 2 presents better performance than version 1 in the single-document evaluation, only the former is evaluated in this experiment. Similarly to the single-document summarization, in this experiment a wide variety of ROUGE scores and summaries were generated.

In the DUC 2001 corpus, the methods sentence resemblance to the title, sentence position version 2, sentence centrality cosine, bushy path, and aggregate similarity achieve the top-5 best performance based on R-1. No statistical difference among each technique is observed at 95% of confidence level. The top performing technique in terms of R-2 is the sentence resemblance to the title. It presents statistically similar results to sentence centrality cosine, aggregate similarity, Bushy Path, sentence position version 2, and word frequency. In the DUC 2002 dataset, the methods sentence position version 2 and sentence centrality cosine present the best results based on R-1 and R-2 respectively. There is no statistical difference between them compared with textrank, bushy path, sentence resemblance to the title, and aggregate similarity regarding R-1 and R-2.

In the DUC 2003 corpus, the method sentence centrality cosine achieves the best performance based on R-1, but it does not show significant improvement over aggregate similarity, bushy path, sentence resemblance to the title, sentence position version 2, textrank, TF-ISF, and word frequency. Taking into consideration the R-2 score, the top-2 performing techniques are aggregate similarity and sentence centrality cosine. Aggregate similarity shows a significant improvement over all other techniques, except for the sentence centrality cosine. Regarding the DUC 2004 dataset, bushy path presents the best results based on R-1 and R-2. It shows statistical similar performance to sentence centrality cosine, aggregate similarity, textrank, word frequency, sentence position version 2, TF-ISF, and sentence resemblance to the title in both ROUGE scores.

Centrality measures such as aggregate similarity, bushy path, sentence centrality cosine achieve the best performance in the multi-document task compared with the other methods. This result is reasonable due to the high degree of redundancy among the documents in the same cluster. The method Sentence position version 2 is not as effective as in the single-document task, but it still presents good results. Sentence resemblance to the title also shows good performance, demonstrating the importance of the title to guide the summarization task. Other techniques that measure the importance of words such as word frequency, TF-ISF, and textrank also achieve a reasonable performance.

The performance of each algorithm is dependent on the task (single- or multi-document) and the corpus used. For instance, in the case of single-document summarization, the position and resemblance with the title are the most efficient aspects, whereas in the multi-document task centrality and position are more effective. It can be observed in both tasks that some techniques that show good overall performance in all corpora: aggregate similarity, bushy path, sentence centrality, sentence centrality cosine, sentence position version 2, sentence resemblance to the title, textrank, TF-ISF, and word frequency. Such techniques are considered the best sentence scoring methods in the experiments developed.

4.4. Evaluating techniques for ensembled scoring

This experiment aims at evaluating four strategies to combine the top-10 sentence salience scoring techniques, regarding the R-1 recall, established in the previous section. R-1 recall was adopted due to the good performance it presented during the training phase in recent works (Hong, Marcus, & Nenkova, 2015; Sipos, Shivaswamy, & Joachims, 2012). The chosen techniques are used to generate several combinations and to evaluate different ensemble strategies.

A wide variety of summaries were generated in the previous experiments using each technique and, based on this fact, the hypothesis formulated here is that they can provide a complementary point of view for each other. In other words, the target is to investigate how these techniques can be combined to provide improved results. Four ensemble strategies were assessed:

- **Averaged combination**: In this combination, the salience score of a sentence is given by the average of **N** individual score techniques under consideration.
- **Weighted average combination**: This ensemble strategy associates a weight with each technique involved. Searching the optimal weight value for each technique is an optimization problem. To address this issue, a genetic algorithm (Goldberg, 1989) was applied. Based on previous works on summarization (Abuobieda et al., 2012; Abuobieda et al., 2013), the population size for each document or cluster and the maximum number of generations were defined as 50 and 100, respectively. The R-1 recall was adopted as the fitness function, employing three evolutionary operators: selection, crossover, and mutation to generate the new populations. A k-fold cross validation approach was adopted with k = 10. In the single-document task the documents were randomly divided in 10 folders (k = 10), whereas in the multi-document task each cluster is used as a folder (k = number of clusters). Thus, **k - 1** folders are used as a training set to find the best configuration of the weights, and such best configuration is applied in the folder not selected (testing set). This process is repeated until all the folders are selected as a testing set.
- **Voting-based combination**: In this ensemble method, a set of T techniques is considered as a voter. The top-ranked sentences selected by each technique $t \in T$ to compose the summary receive a vote. The sentences with the highest number of votes are used to compose the final summary. If two sentences have the same number of votes, the preference is given to sentences at the beginning of the document.
- **Condorcet ranking** (Palshikar, Deshpande, & Athiappan, 2012): This method computes the votes of each technique t to decide the winner among S candidate sentences taking into consideration the position of the sentence into the ranking of sentences candidate selected. For this purpose, a matrix M SxS is created and used to compare every sentence s_i (row in M) against every other sentence s_j (column in M). If a sentence s_i is selected by a technique t with a higher score than a sentence s_j, then s_i receives a vote and the entry $M[i, j]$ is updated. After this voting process, the sentences with the highest number of winning votes are selected to be included into the summary.

All possible linear combinations of the top-10 scoring techniques identified in the previous experiment (Section 4.3) based on R-1 recall are tested. For each generated combination, the selected techniques using the ensemble strategies mentioned above were assessed. Finally, the possibility to improve the results considering different weights for each technique is also investigated and, for that, genetic algorithms were applied to find the optimal weight setting for the top-2 best averaged combinations on each corpus.

Single-document summarization

Table 4 reports on the top-2 best performing results for each ensemble strategy on single-document summarization task. Due to space restrictions the abbreviations introduced in Sections 3 and 4.2 are used here. Table 5 shows the results of these combinations in terms of ROUGE-1 recall (R-1), ROUGE-2 recall, and Direct Matching (DM). All top-2 combinations found present similar performance in all corpora.

Tested on the CNN corpus, the Averaged CombA and the Weighted Averaged CombA achieve the top performance based on R-1 and R-2 respectively. They significantly outperform the Condorcet, and the Voting-based combinations, but do not present a significant improvement over the Averaged CombB and the Weighted Averaged CombB, on both ROUGE scores. The Weighted Averaged CombA also achieves the best DM performance. Regarding the DUC 01 dataset, the best combination in both R-1 and R-2 recall is achieved by the Weighted Averaged CombA. It shows a statistical improvement over the Condorcet combinations, and the Voting-based CombB, in terms of R-1. Based on R-2, the Weighted Averaged CombA provides a significant improvement over the Averaged CombB and the Condorcet combinations. On the DUC 2002 corpus, the top performing combination is the Weighted Averaged CombA. It significantly outperforms the Condorcet and the Voting-based combinations regarding R-1. Based on R-2, the Weighted Averaged CombB presents the best performance, but it only shows a significant improvement over the Condorcet combinations.

Multi-document summarization

Table 6 presents the top-2 performing combinations for each ensemble method on the multi-document summarization task, while Table 7 shows the performance results of these combinations for each ensemble strategy based on R-1 and R-2 recall. As in the single-document results, in this experiment, the results

Table 4
Top-2 highest combinations for each ensemble strategy on single-document summarization task based on ROUGE-1 recall.

Combinations	CNN
Condorcet CombA	NER, NP_VP, OpenRel, SentCen, SentLen, SentPosV2, SentRST, TextRank, WordFreq
Condorcet CombB	NER, NP_VP, OpenRel, SentCen, SentLen, SentPosV2, SentRST, TextRank, TF-ISF
Averaged CombA	SentCen, SentPosV2, TF-ISF
Averaged CombB	SentCen, SentLen, SentPosV2, WordFreq
Voting-based CombA	NER, NP_VP, SentCen, SentLen, SentPosV2, SentRST, TextRank, WordFreq
Voting-based CombB	NER, OpenRel, SentCen, SentLen, SentPosV2, SentRST, TextRank, TF-ISF, WordFreq
Combinations	DUC 2001
Condorcet CombA	SentPosV2, WordFreq
Condorcet CombB	BushyPath, OpenRel, SentPosV2, SentRST
Averaged CombA	SentCenCos, SentPosV2
Averaged CombB	BushyPath, SentPosV2
Voting-based CombA	BushyPath, SentPosV2, SentRST
Voting-based CombB	BushyPath, SentPosV2, WordFreq
Combinations	DUC 2002
Condorcet CombA	SentPosV2, WordFreq
Condorcet CombB	SentPosV2, TF-ISF
Averaged CombA	BushyPath, SentPosV2, TF-ISF
Averaged CombB	AggSim, SentPosV2, TF-ISF
Voting-based CombA	BushyPath, SentPosV2, SentRST
Voting-based CombB	AggSim, SentPosV2, SentRST

Table 5
Results (%) and standard deviation (in parentheses) of the top-2 highest combinations for each ensemble strategy on single-document summarization task. The highest performance on each corpus are highlighted in bold and the group of combinations statistically similar is indicated by a †.

Combinations	CNN		
	DM	R-1	R-2
Condorcet CombA	27.47	57.00 (20.37)	39.40 (25.80)
Condorcet CombB	26.77	56.86 (20.20)	38.91 (25.69)
Averaged CombA	29.89	**57.93** (20.22)	41.54† (25.32)
Averaged CombB	28.39	57.87† (20.18)	40.60† (25.63)
Voting-based CombA	27.76	56.98 (20.42)	39.63 (25.75)
Voting-based CombB	27.92	56.95 (20.27)	39.63 (25.55)
Weighted Avg. CombA	**30.05**	57.86† (20.20)	**41.68** (25.17)
Weighted Avg. CombB	29.53	57.89† (20.34)	41.53† (25.41)
Combinations	DUC 2001		
	DM	R-1	R-2
Condorcet CombA	-	43.70 (10.32)	19.50 (11.60)
Condorcet CombB	-	44.06 (9.78)	19.40 (11.20)
Averaged CombA	-	44.82† (9.53)	19.72† (11.09)
Averaged CombB	-	44.57† (9.74)	19.40 (11.20)
Voting-based CombA	-	44.28† (10.08)	19.72† (11.50)
Voting-based CombB	-	44.30 (10.48)	19.78† (11.88)
Weighted Avg. CombA	-	**44.98** (9.56)	**20.12** (11.04)
Weighted Avg. CombB	-	44.72† (9.51)	19.72† (10.93)
Combinations	DUC 2002		
	DM	R-1	R-2
Condorcet CombA	-	46.66 (9.20)	21.89 (9.87)
Condorcet CombB	-	46.63 (9.28)	21.90 (9.96)
Averaged CombA	-	47.55† (8.60)	22.06† (9.74)
Averaged CombB	-	47.53† (8.70)	22.13† (9.87)
Voting-based CombA	-	47.22 (9.38)	22.17† (10.32)
Voting-based CombB	-	47.19 (9.25)	22.14† (10.22)
Weighted Avg. CombA	-	**47.73** (8.47)	22.34† (9.56)
Weighted Avg. CombB	-	47.64† (8.59)	**22.39** (9.74)

obtained by the top-2 best combination also present close results, mainly in the R-2 recall.

The Weighted Averaged CombA achieves the best performance based on R-1 and R-2 for the DUC 01 corpus. It shows a significant improvement over the averaged and voting-based combinations taking into consideration R-1. The results of all combinations in terms of R-2 are very close; thus, no statistical difference among them is observed.

On the DUC 02 dataset, the Averaged CombA is the top performing combination based on R-1. It significantly outperforms the Condorcet CombB, the Voting-based CombA, and the Weighted Averaged CombB. Regarding R-2, the Condorcet CombA and the Voting-based CombB achieve the best results, showing a significant improvement over the Condorcet CombB, and the Voting-based CombA.

Concerning both the R1 and R2 results for the DUC 03 corpus, the top performing combination is the Weighted Averaged CombB. It largely outperforms the Condorcet CombB, the Voting-based CombB, and the Weighted Averaged CombA regarding R-1; and only over the Voting-based combinations based on R-2.

The Averaged CombA shows the best performance according to R-1 and R-2 in the DUC 04 dataset. It only significantly outperforms the Weighted Averaged CombB based on R-1. The results in terms of the R-2 for all combinations are very close, thus, there is no statistical difference among them.

The average combination strategy presents better performance than the Condorcet and the Voting-based methods in almost all contexts. Besides, it requires fewer techniques to be computed to achieve good performance than the Condorcet and the Voting-based methods. Surprisingly, the approach of considering different weights in the averaged combination by means of a genetic algorithm is not as effective as expected in the experiments made. Although it leads to an improvement in most of the cases, only in the DUC 2001 multi-document task is a significant improvement over the averaged combinations at 95% of confidence level observed.

A wide variety of combinations achieve the top performance in each corpus. No combination achieved the best performance in more than one corpus. This fact suggests that there are features of the corpora that impact in the performance of the investigated techniques and combinations. Identifying and exploiting such characteristics to better select the most suitable methods or their combinations to summarize the document may be an interesting direction to further improve the performance of ATS systems.

Although different combinations have obtained the best performance in each corpus, one can observe the techniques that

Table 6
Top-2 highest combinations for each ensemble strategy on multi-document summarization task based on ROUGE-1 recall.

Combinations	DUC 2001	DUC 2002
Condorcet CombA	SentCenCos, SentPosV2	SentCenCos, SentPosV2, TextRank
Condorcet CombB	SentPosV2, TextRank	AggSim, SentPosV2
Averaged CombA	AggSim, CuePhr, SentPosV2, TextRank	AggSim, BushyPath, SentCenCos, SentPosV2, SentRST, WordFreq
Averaged CombB	BushyPath, SentPosV2, TextRank	AggSim, SentPosV2, TextRank
Voting-based CombA	CuePhr, SentCenCos, SentPosV2	BushyPath, SentPosV2
Voting-based CombB	AggSim, OpenRel, SentPosV2	SentCenCos, SentPosV2
Combinations	**DUC 2003**	**DUC 2004**
Condorcet CombA	AggSim, BushyPath, SentPosV2, TextRank	AggSim, OpenRel, SentCenCos, SentPosV2, TF-ISF
Condorcet CombB	AggSim, SentCenCos, SentPosV2, TextRank	BushyPath, OpenRel, SentCenCos, SentPosV2, TF-ISF
Averaged CombA	AggSim, SentCenCos, TextRank	AggSim, BushyPath, NER, SentPosV2, TextRank
Averaged CombB	AggSim, TextRank	AggSim, NER, SentCenCos, SentPosV2, TextRank
Voting-based CombA	AggSim, BushyPath, NER, SentPosV2, TextRank	BushyPath, OpenRel, SentCenCos, SentPosV2, SentRST, TF-ISF
Voting-based CombB	AggSim, BushyPath, WordFreq	SentCenCos, SentPosV2, TextRank

Table 7
Results (%) and standard deviation (in parentheses) of the Top-2 highest combinations for each ensemble strategy on multi-document summarization task. The highest performance for each corpus is marked in bold and the group of combinations statistically similar is indicated by a †.

Combinations	DUC 2001		DUC 2002	
	R-1	R-2	R-1	R-2
Condorcet CombA	32.29† (6.42)	7.12† (5.77)	35.67† (5.20)	**8.04** (3.56)
Condorcet CombB	32.32† (7.11)	6.98† (6.00)	33.82 (5.04)	7.27 (3.42)
Averaged CombA	32.45 (8.20)	7.51† (6.04)	**35.83** (5.06)	7.83† (3.59)
Averaged CombB	32.55 (7.14)	6.91† (4.87)	34.99† (5.06)	8.02† (3.93)
Voting-based CombA	31.89 (7.07)	7.00† (6.18)	33.87 (5.04)	7.27 (3.42)
Voting-based CombB	31.55 (6.43)	7.16† (5.99)	5.70† (4.97)	**8.04** (3.40)
Weighted Avg. CombA	**33.63** (7.79)	**7.67** (6.10)	35.55† (5.34)	7.80† (3.58)
Weighted Avg. CombB	32.96† (7.84)	7.45† (6.04)	34.73 (5.07)	7.93† (3.45)
Combinations	**DUC 2003**		**DUC 2004**	
	R-1	R-2	R-1	R-2
Condorcet CombA	39.20† (7.12)	8.96† (4.52)	38.06† (4.88)	9.34† (3.23)
Condorcet CombB	38.41 (6.84)	9.42† (4.71)	38.24† (4.49)	9.46† (3.25)
Averaged CombA	39.10† (7.98)	9.28† (5.18)	**38.58** (4.23)	**9.80** (2.98)
Averaged CombB	38.68† (7.74)	8.94† (4.71)	38.50† (4.18)	9.70† (3.04)
Voting-based CombA	38.84† (7.31)	8.86 (4.97)	38.37† (4.27)	9.43†(2.91)
Voting-based CombB	38.40 (6.84)	8.88 (4.85)	37.81† (4.48)	9.48† (3.18)
Weighted Avg. CombA	38.52 (9.17)	9.43† (5.84)	38.15† (4.29)	9.58† (3.02)
Weighted Avg. CombB	**39.61** (9.14)	**9.76** (5.39)	37.68 (4.33)	9.52† (2.82)

are most adopted and their recurrent associations. For instance, the top-6 most used techniques in the top-ranked combinations at both single- and multi-document summarization are Aggregate Similarity, Bushy Path, Sentence Centrality Cosine, Sentence Position version 2, Sentence Resemblance Title, and TextRank. In the single-document task, the co-occurrence between Sentence Position version 2 and Sentence Resemblance Title is the most frequent, whereas in the multi-document summarization is the association between Sentence Position version 2, TextRank, and one centrality measure such as Aggregate Similarity, Bushy Path, or Sentence Centrality Cosine.

4.5. Evaluating a strategy for ML-based summarization

This experiment assesses the performance of the sentence salience scoring techniques presented in Section 3, as input features to various Machine Learning (ML) algorithms. Ten ML algorithms available at the Weka Toolkit are assessed: *AdaBoostM1* (Freund & Schapire, 1996), *J48* (Quinlan, 1993), K-nearest Neighbours (Aha, Kibler, & Albert, 1991) referred as *IBK, Multilayer Perceptron* (Haykin, 1998), Multinomial Logistic Regression (Logistic) (Le Cessie & Van Houwelingen, 1992), *Naive Bayes* (John & Langley, 1995), *Random Forest* (Breiman, 2001), *Random Tree* (Breiman, 2001; Quinlan, 1992), Radial Basis Function Network (*RBFNetwork*) (Broomhead & Lowe, 1988), and Support Vector Machines using Sequential Minimal Optimization (*SMO*) (Platt, 1998).

These algorithms were chosen due to their popularity and because they represent several categories of ML approaches. The goal here is two-fold: (i) identifying the best ML algorithm applied to the sentence salience scoring classification task in each corpus, and (ii) finding the set of the most relevant features according to a feature selection algorithm. Accordingly, each ML algorithm is assessed under two settings: (i) transforming all scoring techniques investigated so far as input features; and (ii) reducing the number of the input features (dimensionality reduction) by applying the Correlation-based Feature Subset Selection (CFS) algorithm.

This experiment addresses the sentence extraction task as a two-class classification problem. The ML algorithms are used to classify the sentences of a document into two classes (IN-SUMMARY and NON-SUMMARY). An important problem to be solved is the unbalance between the examples of classes on the training data. Such problem occurs because the models of the summaries have a higher compression rate; thus, there are far more sentences not in the summaries (NON-SUMMARY). To address such a problem, the oversampling technique was used to duplicate the examples of the minority class (IN-SUMMARY) until reaching the same number of examples of the majority class (NON-SUMMARY). This strategy presented better results in most of the cases of the experiments performed than applying the

Table 8
Results (%) and standard deviation (in parentheses) of the machine learning algorithms evaluation on single-document summarization task. The highest performance on each corpus are highlighted in bold and the group of algorithms statistically similar is indicated by a †.

ML Algorithms	CNN			DUC 2001		DUC 2002	
	DM	R-1	R-2	R-1	R-2	R-1	R-2
AdaBoostM1	31.14	53.29 (20.75)	39.21 (24.40)	44.33† (9.65)	19.54† (10.98)	47.24 (8.93)	22.39† (9.63)
AdaBoostM1_CFS	31.18	53.70 (20.59)	39.50 (24.45)	44.38† (9.96)	19.66† (11.28)	47.37† (9.06)	22.52† (9.69)
IBK	20.60	44.22 (19.01)	27.14 (22.13)	43.18 (10.00)	18.10 (11.25)	46.22 (9.25)	21.01 (9.96)
IBK_CFS	21.29	44.75 (19.01)	28.06 (22.39)	42.76 (10.11)	17.63 (11.28)	45.56 (9.08)	20.55 (9.73)
J48	17.15	38.93 (19.33)	23.89 (21.43)	37.73 (11.77)	15.26 (10.24)	40.69 (11.79)	18.05 (9.74)
J48_CFS	17.92	40.11 (19.35)	24.76 (21.50)	37.70 (11.05)	15.09 (9.35)	41.77 (10.35)	18.47 (9.36)
Logistic	**32.54**	55.83† (20.78)	41.58† (24.65)	**44.65** (9.78)	19.82† (11.16)	**47.76** (8.93)	**22.67** (9.72
Logistic_CFS	32.17	56.04† (20.62)	**41.67** (24.60)	44.56† (9.79)	19.80† (11.32)	47.38† (8.97)	22.40† (9.61)
Multilayer Perceptron	31.67	54.27 (20.76)	39.91 (24.75)	43.81 (8.97)	18.93 (10.32)	47.04 (9.86)	22.19† (10.47)
Multilayer Perceptron_CFS	32.36	54.94 (20.91)	40.95† (24.87)	44.40† (9.86)	19.82† (11.33)	46.77 (9.28)	21.97 (9.73)
Naive Bayes	28.15	56.17† (20.24)	39.30 (25.30)	42.77 (10.13)	17.96 (11.08)	45.59 (9.57)	20.60 (9.77)
Naive Bayes_CFS	28.13	55.36 (20.24)	38.74 (25.00)	43.18 (9.77)	18.34 (10.98)	45.92 (9.70)	20.98 (9.81)
Random Forest	12.36	32.88 (18.60)	25.74 (21.69)	31.20 (13.41)	14.15 (9.80)	35.75 (13.43)	17.01 (8.92)
Random Forest_CFS	13.68	32.11 (18.78)	24.93 (21.66)	31.94 (13.52)	14.18 (10.50)	36.82 (13.09)	17.37 (9.39)
Random Tree	17.93	38.44 (19.50)	25.19 (21.89)	36.32 (12.77)	15.06 (10.09)	39.31 (12.11)	17.56 (9.62)
Random Tree_CFS	17.66	38.27 (19.41)	24.77 (21.60)	36.45 (11.84)	14.99 (10.13)	39.29 (12.52)	17.72 (9.81)
RBF Network	28.14	**56.22** (20.25)	39.44 (25.41)	42.39 (10.31)	17.38 (11.19)	45.46 (9.25)	20.52 (9.87)
RBF Network_CFS	28.70	55.96† (20.18)	39.51 (25.19)	43.45 (9.74)	18.64 (10.86)	46.46 (8.99)	21.33 (9.51)
SMO	28.58	48.65 (21.68)	35.40 (25.03)	44.48† (9.73)	**19.97** (11.33)	47.22 (8.99)	22.35† (9.80)
SMO_CFS	28.80	48.86 (21.66)	35.66 (24.95)	44.46† (9.79)	19.96† (11.42)	47.29 (8.99)	22.42† (9.82)

Synthetic Minority Over-sampling TEchnique (SMOTE) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002).

In all experiments, the k-fold cross validation was adopted for estimating the performance of the ML algorithms. For each folder, the sentences extraction task is performed in two steps:

- **Training:** In this step, k - 1 folders are used as training set, and a classification model is created. In the single-document task, the documents were split into 10 folders (k = 10), whereas in the multi-document task, each cluster forms a folder;
- **Testing:** The folder not selected in the training step is used as a test set. The model created in the previous step is applied to classify the sentences. For each sentence, the algorithm gives a confidence score, which is employed for ranking the sentences classified as IN-SUMMARY. This way, only the sentences with the highest confidence score are selected under the constraint of the compression rate.

The CNN summary models are extractive; therefore, they can be directly used as training examples. However, since the DUC models are abstractive, some adjustments are needed. Each sentence of the human model available is mapped onto the most similar sentence of the document. The cosine similarity measure is used to identify the most similar sentences. The mapped sentences are used as positive examples (IN-SUMMARY class) and the other sentences as negative examples (NON-SUMMARY class).

Single-document summarization

Table 8 summarizes the performance of the machine learning algorithms in the single-document summarization task on CNN, DUC 2001, and DUC 2002 datasets. For the CNN corpus, the top performing algorithm based on R-1 is the RBFNetwork. However, it does not present significant improvements over Naive Bayes, RBFNetwork_CFS, Logistic_CFS, and Logistic algorithms. Based on R-2, the top-2 results are achieved by Logistic_CFS and Multilayer Perceptron_CFS, no significant difference between them is observed. The Logistic algorithm reaches the highest DM score, indicating that this algorithm provides the best accuracy to identity the best sentence candidates for the summary.

For the DUC 2001 dataset, Logistic achieves the best results based on R-1. It presents statistical similar results with AdaBoostM1, AdaBoostM1_CFS, Logistic_CFS, Multilayer Perceptron_CFS, SMO, and SMO_CFS. Based on R-2, SMO is the top performing algorithm. It does not present a significant improvement over AdaBoostM1, AdaBoostM1_CFS, Logistic, Logistic_CFS, Multilayer Perceptron_CFS, and SMO_CFS.

On the DUC 2002 corpus, Logistic achieves the best performance on both R-1 and R-2. Based on R-1, it significantly outperforms all other algorithms, except for AdaBoostM1_CFS. Regarding R-2, it presents statistically similar results to AdaBoostM1, AdaBoostM1_CFS, Logistic_CFS, Multilayer Perceptron, SMO, and SMO_CFS.

Multi-document summarization

Table 9 reports the performance of the machine learning algorithms in the multi-document summarization task using the DUC 2001–2004 datasets. On the DUC 2001 corpus, SMO reaches the top performance based on both R-1 and R-2. It does not present any statistical difference over SMO_CFS, AdaBoostM1, AdaBoostM1_CFS, and RandomTree in terms of R-1. Taking into consideration R-2, SMO does not present a significant improvement over AdaBoostM1, AdaBoostM1_CFS, SMO_CFS, and RandomTree.

For the DUC 2002 dataset, AdaBoostM1 and its variation AdaBoostM1_CFS present the best results on R-1 and R-2 respectively. AdaBoostM1 shows statistically similar results based on R-1 with MultilayerPerceptron_CFS, Logistic, Logistic_CFS, RandomTree_CFS, AdaBoostM1_CFS, J48, SMO, and SMO_CFS. Taking into consideration R-2, AdaBoostM1_CFS does not present significant improvement over Logistic, RandomTree_CFS, AdaBoostM1, Logistic_CFS, and J48.

For the DUC 2003 corpus, RBFNetwork_CFS achieves the best results on R-1, whereas Naive Bayes reaches the best performance in terms of R-2. RBFNetwork_CFS shows statistical similar results with Logistic, RBFNetwork, NaiveBayes, and NaiveBayes_CFS. Taking into consideration R-2, Naive Bayes does not present significant improvement over Logistic_CFS, RBFNetwork, Multilayer Perceptron, and Logistic. For the DUC 2004 dataset, the top performing algorithm is the Logistic_CFS in both R-1 and R-2. Based on R-1 and R-2, it presents statistically similar results to Logistic, NaiveBayes, NaiveBayes_CFS, and RBFNetwork_CFS.

Table 9
Results (%) and standard deviation (in parentheses) of the machine learning evaluation on multi-document summarization task. The top performance for each corpus is marked in boldface and the group of algorithms statistically similar is indicated by a †.

ML Algorithms	DUC 2001		DUC 2002		DUC 2003		DUC 2004	
	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
AdaBoostM1	31.70† (6.61)	6.58† (5.47)	**34.17** (5.29)	7.58† (3.92)	35.06 (4.70)	7.62 (3.24)	35.78 (4.69)	8.28 (2.61)
AdaBoostM1_CFS	31.97† (6.70)	6.58† (5.93)	33.77† (5.32)	**7.62** (4.31)	35.06 (4.70)	7.72 (3.26)	36.60 (4.94)	8.89 (2.79)
IBk	27.42 (5.60)	3.71 (2.74)	31.18 (5.66)	5.33 (2.66)	35.01 (6.70)	6.32 (3.56)	35.19 (4.41)	7.25 (2.92)
IBk_CFS	28.94 (5.92)	4.32 (2.36)	31.38 (5.47)	5.76 (3.06)	34.86 (7.05)	7.03 (4.03)	34.67 (5.14)	7.49 (3.21)
J48	29.00 (7.11)	4.76 (4.07)	33.77† (6.11)	7.11† (4.78)	34.31 (6.90)	6.78 (4.20)	35.19 (5.14)	7.26 (3.05)
J48_CFS	28.91 (5.74)	3.99 (2.72)	31.02 (4.77)	5.17 (2.59)	31.91 (8.74)	5.86 (3.42)	33.07 (4.34)	6.36 (2.64)
Logistic	28.16 (6.97)	5.02 (4.18)	34.14† (6.39)	7.60† (4.03)	36.56† (6.28)	8.03† (4.58)	37.09† (4.80)	9.02† (3.06)
Logistic_CFS	27.31 (7.61)	4.63 (3.92)	34.08† (6.34)	7.57† (4.00)	35.35 (7.90)	8.19† (4.83)	**37.48** (5.33)	**9.46** (3.21)
MultilayerPerceptron	28.21 (5.57)	4.29 (2.71)	32.05 (6.30)	6.46 (3.27)	35.67 (7.50)	8.12† (4.75)	36.33 (4.98)	8.34 (2.79
MultilayerPerceptron_CFS	28.39 (8.10)	4.42 (3.40)	34.16† (5.90)	7.21 (3.81)	35.27 (6.93)	7.76 (4.95)	36.89 (4.57)	8.66 (2.85)
NaiveBayes	26.67 (7.40)	4.09 (3.78)	31.14 (8.14)	6.02 (3.86)	36.29† (7.60)	**8.45** (5.31)	37.29†4.32)	9.23† (3.04)
NaiveBayes_CFS	27.58 (7.64)	4.63 (4.08)	31.61 (7.92)	6.43 (3.67)	36.23† (7.23)	7.55† (5.25)	37.11† (4.64)	9.24† (3.17)
RandomForest	25.04 (6.31)	5.00 (3.60)	32.99 (6.14)	6.58 (4.09)	18.34 (8.01)	3.73 (2.32)	34.03 (8.10)	8.39 (3.32)
RandomForest_CFS	25.88 (6.83)	4.24 (3.27)	33.46 (5.87)	7.07 (3.14)	22.42 (7.24)	4.69 (1.96)	36.18 (4.63)	8.87 (2.95)
RandomTree	30.74† (7.84)	5.96† (3.88)	33.83 (4.94)	6.79 (3.27)	31.92 (7.80)	6.56 (3.84)	36.80 (4.73)	8.84 (3.04)
RandomTree_CFS	27.69 (5.27)	4.58 (2.76)	34.0† (5.25)	7.60† (3.97)	32.68 (8.07)	6.87 (4.61)	36.26 (5.30)	8.35 (3.43)
RBFNetwork	26.29 (7.03)	3.91 (3.49)	31.19 (8.27)	5.79 (3.75)	36.39† (7.34)	8.18† (5.56)	36.89 (4.52)	8.96 (3.22)
RBFNetwork_CFS	27.70 (7.85)	4.64 (4.08)	31.84 (7.69)	6.48 (3.67)	**37.13** (6.81)	7.93 (5.21)	37.24† (4.69)	9.28† (3.28)
SMO	**32.74** (6.31)	**7.13** (6.02)	33.66† (4.90)	7.05 (3.32)	33.97 (5.12)	7.15 (3.05)	35.40 (4.91)	8.03 (2.59)
SMO_CFS	32.08† (6.59)	6.45† (5.30)	33.66† (4.90)	7.05 (3.32)	34.98 (5.84)	7.64 (4.13)	35.43 (4.89)	8.01 (2.56)

Based on the performed experiments in both single- and multi-document summarization some conclusions may be drawn:

- As expected, the CFS algorithm reduces the dimensionality of the features and, in general, slightly influences R-1 and R-2 for both tasks. In 57.14% of the seventy comparisons in all corpora in both single- and multi-document task the ML algorithm version using CFS improves the results based on R-1. On the other hand, in 95.71% of these cases, wherein the CFS leads to decrease the performance, no significant difference at 95% of confidence level are observed.
- In the single-document summarization task, Logistic_CFS and RBFNetwork_CFS present the top-2 overall performance in terms of R-1 in all corpora. The features selected as relevant by CFS algorithm taking into consideration all corpora are: bushy path, named entities frequency, numerical data, sentence centrality, sentence centrality cosine, sentence position version 2, sentence resemblance to the title, textrank, TF-ISF, and word frequency.
- In the multi-document summarization, AdaBoostM1_CFS and Logistic show the top-2 overall performance based on R-1 in all corpora. The features selected as relevant by CFS algorithm taking into consideration all datasets are: aggregate similarity, lexical similarity, sentence centrality, sentence centrality cosine, sentence position version 2, sentence resemblance to the title, textrank, TF-ISF, word co-ocurrence, and word frequency.
- Algorithms such as Logistic and AdaBoostM1 achieved a high recall to correctly classify the sentences in the SUMMARY and NON-SUMMARY classes. However, the confidence score assigned to the sentences classified into the SUMMARY class, in some cases, did not succeed in distinguishing the best sentences to be included in the summary. Therefore, there was a corresponding degradation in the performance of those algorithms according to the DM and ROUGE measures.

4.6. Comparative evaluation with the state-of-the-art

This section compares the performance of the top performing individual sentence scoring technique (Section 4.3), the best ensemble strategy (Section 4.4), and the top ML algorithm (Section 4.5), against several state-of-the-art summarizers.

Single-document summarization.

In the single-document task, the summarization results were compared with **(i)** the best performing participants at the DUC 2001 and 2002 shared-task found by the experiments presented here; and **(ii)** the following three top performing summarizers according to a previous evaluation reported in Batista et al. (2015):

- **Autosummarizer** (Autosummarizer, 2015) consists of a web service that produces a summary by splitting and ranking the most relevant sentences. Its single-document summarization algorithm extracts the most important sentences from the original document, reaching good performance in a previous assessment against different summarizers (Batista et al., 2015). Unfortunately, details of how this system works are not available.
- **Classifier4J** (Classifier4J, 2005) is a toolkit for text classification and summarization. It performs an extractive single-document summarization based on word frequency. Classifier4J selects the first sentences containing one of the top-100 most frequent words in the document to compose the summary.
- **HP-UFPE Functional summarization** (HP-UFPE FS) (Ferreira et al., 2014; 2013) is a summarization system based on the seventeen extractive summarization strategies more widely acknowledged in the literature, and that are extensively evaluated on news articles, blogs, and scientific documents. This work makes use of the HP-UFPE FS system adopting the best sentence scoring combination for news articles reported in Ferreira et al. (2014). Such features are: TF-ISF, Lexical Similarity, Sentence Position version 1, and Sentence resemblance to the title.

Table 10 shows the comparative results of the best performing setting found in the experiments against the aforementioned summarizers based on ROUGE-1 (R-1) recall, ROUGE-2 (R-2) recall, and Directing Matching (DM) on single-document summarization task.

Starting with the CNN dataset, the best combination (Averaged CombA) achieves the best performance in terms of R-1, R-2, and DM scores. It statistically outperforms all other systems at 95% of confidence level. The best ML algorithm (RBFNetwork) and the best individual technique (TF-ISF) also present significantly better performance than the three related work summarizers. Regarding the DM, the results demonstrate that there is plenty of room for improvements. The best results found in this experiment

Table 10
Systems performance (%) and standard deviation (in parentheses) on single-document summarization task. The highest performance on each corpus are highlighted in bold and the group of summarizers statistically similar is indicated by a †.

Systems	CNN		
	DM	R-1	R-2
AutoSummarizer	23.16	48.81 (18.70)	32.74 (22.70)
Averaged CombA	**29.89**	**57.93** (20.22)	**41.54** (25.32)
Classifier4J	23.89	46.63 (20.32)	32.15 (23.13)
HP-UFPE FS	24.75	50.71 (20.34)	34.58 (24.38)
RBFNetwork	28.14	56.22 (20.25)	39.44 (25.41)
TF-ISF	23.99	54.25 (19.87)	35.65 (25.44)
Systems	DUC 2001		
	DM	R-1	R-2
AutoSummarizer	-	41.92 (9.04)	16.63 (9.95)
Classifier4J	-	44.44† (9.85)	19.86† (11.34)
HP-UFPE FS	-	35.91 (11.78)	11.78 (9.78)
Logistic	-	44.65† (9.78)	19.82† (11.16)
Sentence Position version 2	-	43.75 (10.47)	19.57† (11.64)
System T[a]	-	44.53† (9.23)	**20.27** (10.75)
Weighted Averaged CombA	-	**44.98** (9.56)	20.12† (11.04)
Systems	DUC 2002		
	DM	R-1	R-2
AutoSummarizer	-	43.79 (8.78)	19.17 (9.31)
Classifier4J	-	47.09 (8.93)	22.12 (9.87)
Logistic	-	47.76† (8.93)	22.67† (9.72)
HP-UFPE FS	-	45.70 (9.31)	20.59 (9.88)
Sentence Position version 2	-	46.94 (9.20)	22.14 (10.01)
System 28	-	**48.07** (8.90)	**22.88** (9.96)
Weighted Averaged CombA	-	47.73† (8.47)	*22.34*† (9.56)

[a] It was observed that the summaries of the participating systems on DUC 2001 available are incomplete.

were achieved by Averaged CombA, which is below 30% accuracy. Even taking into consideration the best overall DM performance in all experiments, which was reached by the Logistic algorithm (32.54%), it is still a very low performance. It is important to notice that since all decisions were focused on R-1 recall, it is possible that other combinations not considered into the top-2 combinations, but with a high performance on ROUGE precision can also present improved results based on the DM score.

Concerning the R-1 results on the DUC 01 corpus, the best combination (Weighted Averaged CombA) obtained the top performing result, but it is not statistically different from the one presented by Classifier4J, the best machine learning algorithm (Logistic), and the best DUC participant (System T). In terms of R-2, System T performed best, but it shows a significantly better performance only over AutoSummarizer and HP-UFPE FS.

On the DUC 02 dataset, the top-3 performing systems are the best DUC participant (System 28), the best machine learning algorithm (Logistic), and the best combination (Weighted Averaged CombA) based on both ROUGE scores. There is no significant difference among them on both scores. The Weighted Averaged CombA statistically outperforms the best individual technique (Sentence Position version 2), AutoSummarizer, Classifier4J, and HP-UFPE FS.

Even after more than one decade since the DUC 2001 and 2002 competition, System T and 28 still present very competitive results against more recently proposed summarizers. Applying only shallow sentence scoring techniques, it was not possible to provide significant improvements over them in the performed experiments. The Weighted Averaged CombA on both DUC corpora, significantly outperforms the Sentence Position version 2, which was used as a baseline in the original DUC competition, and present competitive results compared with AutoSummarizer, Classifier4J, and HP-UFPE FS.

Fig. 1 presents in a tabular form further details of the statistical significance analysis of the ROUGE-1 recall distributions on all the three corpora. The p-values of the corresponding statistical tests performed between the summarizers compared are presented in different hues of gray. A low p-value, near to 0.0 and 0.2, indicates a significant difference in the performance with a confidence level of 99% and 95%, respectively.

Table 11 shows three examples of summaries produced by the System 28, the Weighted Averaged CombA, and the Classifier4J for the document AP880622-0184 of the DUC 2002 corpus. This article reports on a charitable concert scheduled to celebrate Leonard Bernstein's 70th birthday. The three systems generated similar summaries and presented most of the information in the human-made reference summaries. The three systems achieved high ROUGE-1 recall scores: the Weighted Averaged CombA (70.79%), Classifier4J (70.79%), and System 28 (61.38%).

Multi-document summarization.

The results obtained for the multi-document task are compared with the **(i)** best performing system participant at the DUC 2001–2004 competition found in the experiments here, and **(ii)** the following state-of-the-art systems: ICSISumm (Gillick & Favre, 2009), Greedy-KL (Hong et al., 2015), LLRSum (Conroy, Schlesinger, & O'Leary, 2006), ProbSum (Nenkova, Vanderwende, & McKeown, 2006), and Sume (Boudin, Mougard, & Favre, 2015). The summaries of the Sume system were generated using the original implementation provided by the authors at https://github.com/boudinfl/sume. For the other systems, the summaries were generated and provided by Hong et al. (2015). The results of these comparisons are shown in Table 12.

Regarding R-1 on the DUC 01 corpus, the top-2 performing systems are ICSISumm and the best combination (Weighted Averaged CombA). They do not achieve significant difference over Greedy-KL, LLRSum, the best machine learning algorithm (SMO), and the Sume system. Only the ICSISumm system significantly outperforms the best DUC participant (System P). The Weighted Averaged CombA presents a significantly better performance over ProbSum and the best individual technique (Sentence Resemblance Title). In terms of R-2, ICSISumm achieves the best results, but it only presents a significant improvement over ProbSum and Sentence Resemblance Title.

For the DUC 02 dataset, ICSISumm reaches the highest performance based on both R-1 and R-2, showing a significant improvement over all other systems. Based on R-1, the best combination (Averaged CombA) significantly outperforms the LLRSum, ProbSum, Sume, the best individual technique (Sentence Position version 2), and the best machine learning algorithm (AdaBoostM1_CFS). Only the ICSISumm performs significantly better than the best DUC participant (System 26).

In the case of the DUC 03 corpus, ICSISumm also achieves the highest results based on R-1 and R-2. However, based on R-1, it does not show a statistical improvement over the best combination (Weighted Averaged CombB), the best DUC participant (System 12), Greedy-KL, and Sume system. The Weighted Averaged CombB significantly outperforms, based on R-1, the best individual technique (Sentence Centrality Cosine), the best machine learning algorithm (RBFNetwork_CFS), LLRSum, and ProbSum. None of the evaluated systems statistically outperforms the System 12. In terms of R-2, ICSISumm presents statistically similar performance to the Sentence Centrality Cosine, Weighted Averaged CombB, and System 12.

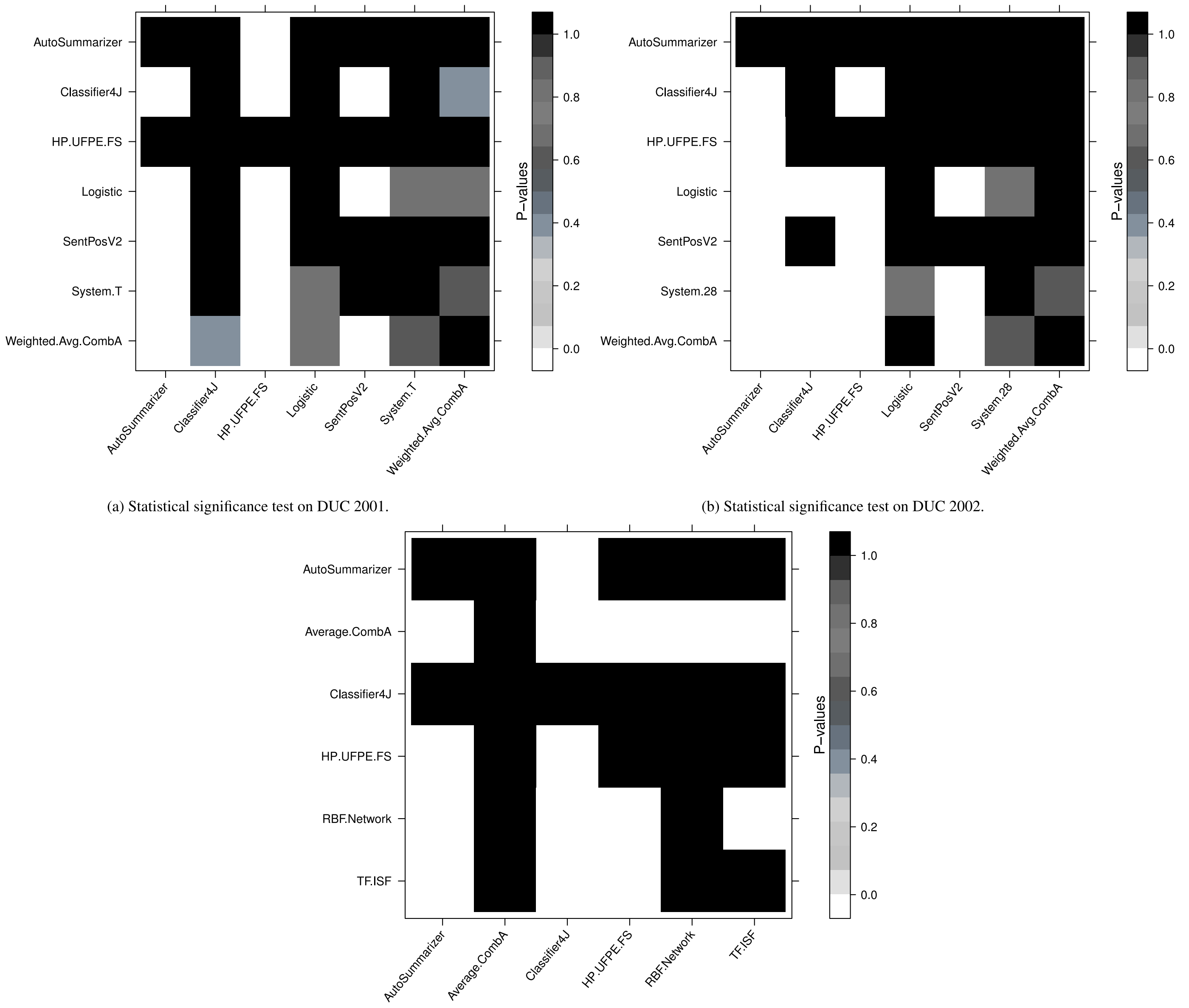
For the DUC 04 dataset, the top performing system is the best combination (Averaged CombA) based on both R-1 and R-2. ICSISumm also presents the best results in terms of R-2. Based on R-1, the Averaged CombA significantly outperforms the best individual technique (Bushy Path), LLRSum, and ProbSum at 99%

Table 11
Summaries generated for the DUC 2002 corpus (document AP880622-0184) by the best DUC participant (System 28), the Weighted Averaged CombA, and the Classifier4J system. The human-made reference summaries provided by the DUC conference are also illustrated.

Systems	Summaries
Classifier4J	Beverly Sills, Lauren Bacall, Betty Comden and Phyllis Newman are among performers who will sing, act and make guest appearances at a birthday bash in August for conductor Leonard Bernstein. The Leonard Bernstein Gala Birthday Performance is a benefit concert scheduled for the composer's 70th birthday, Aug. 25, to raise money for the Tanglewood Music Center, where Bernstein got his conducting start. Sills will be host. Performances will include the Boston Symphony Orchestra, the Boston Pops Orchestra and the Tanglewood Festival Chorus under the direction of some of the many conductors whose careers have been guided by Bernstein.
System 28	Beverly Sills, Lauren Bacall, Betty Comden and Phyllis Newman are among performers who will sing, act and make guest appearances at a birthday bash in August for conductor Leonard Bernstein. The Leonard Bernstein Gala Birthday Performance is a benefit concert scheduled for the composer's 70th birthday, Aug. 25, to raise money for the Tanglewood Music Center, where Bernstein got his conducting start. The concert will celebrate Bernstein's accomplishments in popular music with excerpts from “West Side Story”, “On the Town” and others. Dame Gwyneth Jones and Frederica von Stade will be among those performing highlights from “Fidelio”, “Tristan und Isolde” and other works to honor Bernstein's landmark opera recordings.
Weighted Averaged CombA	Beverly Sills, Lauren Bacall, Betty Comden and Phyllis Newman are among performers who will sing, act and make guest appearances at a birthday bash in August for conductor Leonard Bernstein. The Leonard Bernstein Gala Birthday Performance is a benefit concert scheduled for the composer's 70th birthday, Aug. 25, to raise money for the Tanglewood Music Center, where Bernstein got his conducting start. Performances will include the Boston Symphony Orchestra, the Boston Pops Orchestra and the Tanglewood Festival Chorus under the direction of some of the many conductors whose careers have been guided by Bernstein. Bacall and soprano Barbara Hendricks will perform a movement from Bernstein's Symphony No. 3, “Kaddish”.
Human-made Reference Summary 1	The Leonard Bernstein Gala Birthday Performance is a benefit concert scheduled for the composer's 70th birthday, August 25, to raise money for the Tanglewood Music Center, where Bernstein started his conducting career. Beverly Sills will host. Lauren Bacall, Betty Comden, and Phyllis Newman are among the performers who will sing, act, and make guest appearances. The Boston Symphony Orchestra, the Boston Pops Orchestra, and the Tanglewood Festival Chorus, directed by some of the conductors mentored by Bernstein, will also perform. The concert will celebrate Bernstein's contributions to classical and popular compositions and landmark opera recordings. Tickets cost 20*to*5000.
Human-made Reference Summary 2	Beverly Sills, Lauren Bacall, Betty Comden, and Phyllis Newman will appear at the Leonard Bernstein Gala Birthday Performance in August. The concerts mark his 70th birthday and will benefit the Tanglewood Music Center, where he got his start as a conductor. Conductors who were mentored by Bernstein will direct the Boston Symphony, the Boston Pops Orchestra, and the Tanglewood Festival Chorus. Bernstein's compositions will be honored by selections from “Kaddish”, “Serenade”, “On the Town”, and “West Side Story.” Bernstein's landmark opera recordings will also be honored. Tickets go from $20 on the lawn to $5,000 for benefactors.

Table 12
Systems performance (%) and standard deviation (in parentheses) on multi-document summarization task. The highest performance for each corpus is printed in boldface and the group of summarizers statistically similar is indicated by a †.

Systems	DUC 2001		DUC 2002	
	R-1	R-2	R-1	R-2
Best Combination	33.63† (7.79)	7.67† (6.10)	35.83 (5.06)	7.83 (3.59)
DUC Participant	31.69 (6.43)	6.30† (3.76)	35.21 (5.30)	7.66 (3.30)
Best Individual Technique	30.63 (6.66)	5.94 (3.74)	33.78 (5.65)	7.16 (3.69)
Best ML Algorithm	32.74† (6.31)	7.13† (6.02)	34.17 (5.29)	7.58 (3.92)
Greedy-KL	32.84† (6.43)	6.70† (3.64)	35.79 (5.74)	7.49 (3.61)
ICSISumm	**33.88** (6.95)	**7.75** (4.30)	**37.34** (5.05)	**9.53** (3.83)
LLRSum	32.00† (5.88)	6.76† (3.25)	32.84 (5.55)	6.75 (3.72)
ProbSum	29.73 (5.41)	5.16 (2.64)	32.57 (4.74)	7.06 (3.63)
Sume	33.37† (7.14)	7.73 (4.29)	34.30 (5.26)	8.15 (3.92)
Systems	DUC 2003		DUC 2004	
	R-1	R-2	R-1	R-2
Best Combination	39.61† (9.14)	9.76† (5.39)	**38.58** (4.23)	**9.80** (2.98)
DUC Participant	38.53† (7.98)	9.17† (5.59)	37.69† (4.08)	8.98† (3.08)
Best Individual Technique	37.61 (8.00)	9.42† (5.43)	36.85 (4.97)	8.34 (3.75)
Best ML Algorithm	37.13 (6.81)	7.93 (5.21)	37.48 (5.33)	9.46† (3.21)
Greedy-KL	39.92† (7.82)	8.82 (4.80)	38.27† (4.73)	8.96 (3.09)
ICSISumm	**40.03** (8.05)	**11.06** (6.15)	38.42† (4.14)	**9.80** (3.17)
LLRSum	36.68 (7.74)	7.99 (4.27)	35.90 (5.01)	8.06 (3.12)
ProbSum	36.09 (8.27)	8.94 (3.73)	35.37 (4.41)	8.18 (3.00
Sume	39.34† (7.34)	9.73† (5.51)	37.29 (4.24)	8.83 (2.71)



(a) Statistical significance test on DUC 2001.

(b) Statistical significance test on DUC 2002.

(c) Statistical significance test on CNN corpus.

Fig. 1. Results of the statistical significance tests performed among the summarizers compared in Table 10. The p-values are represented in grayscale. A low *p*-value (0.0 to 0.2 in the grayscale) indicates a significant superiority based on average ROUGE-1 measures among each system (lines) compared with the other systems (columns) with 99% and 95% of confidence level, respectively.

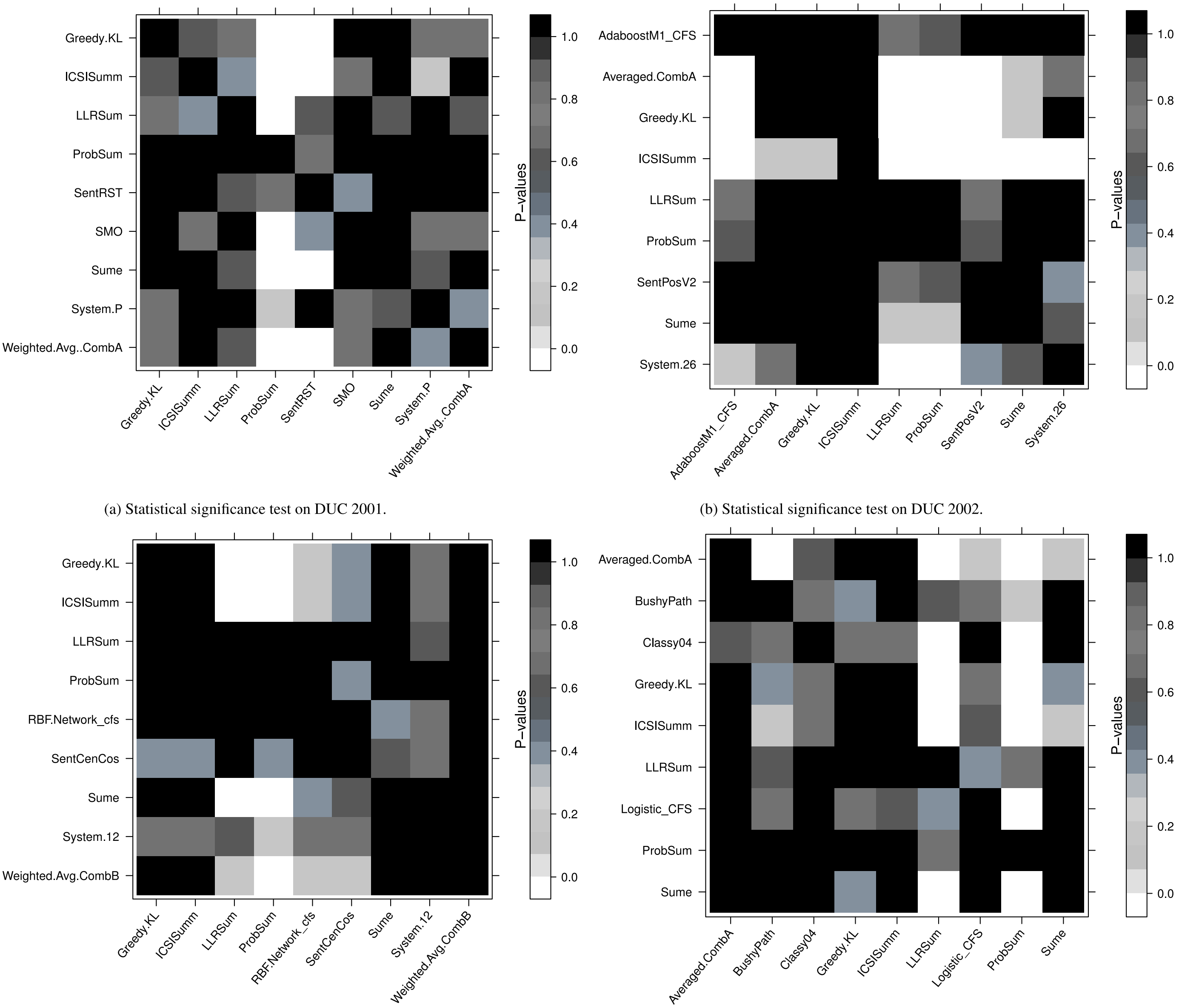
of confidence level; and the best machine learning algorithm (Logistic_CFS), and Sume system at 95% of confidence level. As in DUC 03, no system provided statistical improvement over the best DUC participant CLASSY 04 (System 65) (Conroy, Schlesinger, Goldstein, & Oleary, 2004). In terms of R-2, Averaged CombA and ICSISumm significantly outperform the Bushy Path, Greed-KL, LLRSum, ProbSum, and Sume.

Fig. 2 shows in a tabular form more details on the statistical tests results of the ROUGE-1 recall distributions in all the corpora tested. Among the state-of-the-art systems investigated, ICSISumm shows the best performance, achieving the top results for the DUC 01–03 corpus. This system and the Sume adopt an ILP-based approach (Gillick & Favre, 2009) that optimizes the coverage of the most relevant bigrams of the document(s) into the summary. The relevance of a bigram in these systems is measured by counting the number of documents the bigram is presented. Concept-based ILP approach has been intensively investigated in the literature showing prominent results in the last few years. Despite the advances in multi-document summarization, the performance of the original DUC participants is still very competitive. Only ICSISumm presents a significant improvement over them in the DUC 01–02 based on the figures of R-1, and in DUC 02 in terms of R-2.

The ensemble strategies investigated, in most of the cases, outperformed the results obtained by the individual techniques. In both single- and multi-document summarization tasks, a competitive performance with the state-of-the-art summarizers was achieved through the combination of shallow sentence scoring techniques and applying a simple approach to avoid redundancy.

5. Conclusions and future work

This paper presented an assessment of eighteen sentence salience scoring techniques for both single- and multi-document extractive summarization on news domain. Those scoring tech-



(a) Statistical significance test on DUC 2001.

(b) Statistical significance test on DUC 2002.

(c) Statistical significance test on DUC 2003.

(d) Statistical significance test on DUC 2004.

Fig. 2. Results of the statistical significance tests performed among the summarizers compared in Table 12. The p-values are represented in grayscale. A low *p*-value (0.0 to 0.2 in the grayscale) indicates a significant superiority based on average ROUGE-1 measures among each system (lines) compared with the other systems (columns) with 99% and 95% of confidence level, respectively.

niques were evaluated in many application scenarios: individually, combining, and comparing them with the state-of-the-art summarizers using traditional evaluation measures. In particular, several experiments were conducted on the CNN corpus and the DUC benchmark datasets (2001–2004). The experimental results showed that the effectiveness of the analyzed techniques individually, is reasonable and that the strategy of combining them can lead to significantly improved results. In addition, the experiments demonstrated that the top performing combinations provided competitive results compared with the state-of-the-art summarizers in both single- and multi-document summarization tasks.

The top performing combinations explored the diversity and performance of the best individual methods to optimize the sentence scoring task, leading to improved results. However, no pattern in such combinations to yield the best results in all the corpora tested has been found. The experimental results also showed that the techniques either in isolation or their combinations, or even the related summarizers compared in both single- and multi-document tasks, presented a high standard deviation in all the corpora tested, particularly in the single-document task. This fact demonstrates that even being in the same domain, none of the methods, combinations, and systems evaluated can always achieve a high performance for all documents.

Based on the overall ROUGE-1 recall score in the individual assessment, one can conclude that the top-10 performing scoring techniques in the experiments performed are: aggregate similarity, bushy path, named entities, noun and verbal phrases, sentence centrality cosine, sentence position version 2, Sentence resemblance to the title, textrank, TF-ISF, and word frequency. Regarding the ensemble strategies, the averaged and the weighted averaged combinations showed better performance than the condorcet ranking, voting-based, and machine learning algorithms.

Despite the encouraging results obtained so far, the authors intend to pursue the following lines as further work: **(i)** Investigating what kind of information presented into the human-made reference summaries are not selected by any of the techniques analyzed; **(ii)** Given the good performance achieved by the concept-based ILP systems (ICSISumm and Sume), to explore the feasibility of applying the scoring techniques investigated to estimate the weights of the bigrams adopted in concept-based ILP approaches; **(iii)** Analyzing the feasibility of including a classification step, aiming to estimating the most suitable technique or a combination of techniques to be applied to summarize a document based on different features of the input document or cluster; and **(iv)** Exploring the application of a classification step to discriminate the best document or cluster summary from a set of different summary candidates generated by applying several summarization techniques. These research lines may allow the development of an ATS approach that will be optimized based on characteristics of the input document or cluster following the guidelines presented in Simske (2013).

Acknowledgements

This research was funded by CNPQ-Research Agency (Brazil) and a R&D project between HP Brazil and UFPE originated from tax exemption (IPI - Law nº 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 *Proceedings of the international conference on information retrieval & knowledge management* (pp. 193–197).

Abuobieda, A., Salim, N., Kumar, Y. J., & Osman, A. H. (2013). An improved evolutionary algorithm for extractive text summarization. In *Intelligent information and database systems. Lecture notes in computer science: Vol. 7803* (pp. 78–89). Springer-Verlag Berlin Heidelberg.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning, 6*(1), 37–66.

Autosummarizer (2015). Retrieved from http://autosummarizer.com/.

Banerjee, S., Mitra, P., & Sugiyama, K. (2015). Multi-document abstractive summarization using ILP based multi-sentence compression. In *Proceedings of the 24th international conference on artificial intelligence. IJCAI'15* (pp. 1208–1214). AAAI Press.

Baralis, E., Cagliero, L., Jabeen, S., Fiori, A., & Shah, S. (2013). Multi-document summarization based on the yago ontology. *Expert Systems with Applications, 40*(17), 6976–6984.

Barrera, A., & Verma, R. (2012). Combining syntax and semantics for automatic extractive single-document summarization. In *Proceedings of the conference on intelligent text processing and computational linguistics (CICLING). Lecture Notes in Computer Science: Vol. 7182* (pp. 366–377). Springer.

Batista, J., Ferreira, R., Tomaz, H., Ferreira, R., Dueire Lins, R., Simske, S., et al. (2015). A quantitative and qualitative assessment of automatic text summarization systems. In *Proceedings of the 2015 ACM symposium on document engineering. DocEng '15* (pp. 65–68). New York, NY, USA: ACM.

Binh Tran, G. (2013). Structured summarization for news events. In *Proceedings of the 22nd international conference on world wide web companion* (pp. 343–348). Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.

Binwahlan, M. S., Salim, N., & Suanmali, L. (2009). Swarm based features selection for text summarization. *International Journal of Computer Science and Network Security, 9*(1), 175–179.

Boudin, F., Mougard, H., & Favre, B. (2015). Concept-based summarization using integer linear programming: from concept pruning to multiple optimal solutions. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1914–1918). Lisbon, Portugal: Association for Computational Linguistics.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on world wide web 7. WWW7* (pp. 107–117). Amsterdam, The Netherlands: Elsevier Science Publishers B. V.

Broomhead, D. S., & Lowe, D. (1988). Radial basis functions, multi-variable functional interpolation and adaptive networks. *Complex Systems, 2*, 321–355.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*(1), 321–357.

Christensen, J., Soderland, S., Bansal, G., & Mausam (2014). Hierarchical summarization: scaling up multi-document summarization. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 902–912). Baltimore, Maryland: Association for Computational Linguistics.

Classifier4J (2005). Retrieved from http://classifier4j.sourceforge.net/.

Conroy, J. M., Schlesinger, J. D., Goldstein, J., & Oleary, D. P. (2004). Left--brain/right-brain multi-document summarization. *Proceedings of the Document Understanding Conference (DUC 2004)*.

Conroy, J. M., Schlesinger, J. D., & O'Leary, D. P. (2006). Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL on main conference poster sessions. COLING-ACL '06* (pp. 152–159). Stroudsburg, PA, USA: Association for Computational Linguistics.

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

Etzioni, O., Fader, A., Christensen, J., Soderland, S., & Mausam, M. (2011). Open information extraction: the second generation. In *Proceedings of the twenty-second international joint conference on artificial intelligence. IJCAI'11* (pp. 3–10). AAAI Press.

Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the conference on empirical methods in natural language processing. EMNLP '11* (pp. 1535–1545). Stroudsburg, PA, USA: Association for Computational Linguistics.

Fattah, M. (2014). A hybrid machine learning model for multi-document summarization. *Applied Intelligence, 40*(4), 592–600.

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

Ferreira, R., de Freitas, F. L. G., de Souza Cabral, L., Lins, R. D., Lima, R., de França Pereira e Silva, G., et al. (2014). A context based text summarization system. In *Proceedings of the 11th international workshop on document analysis systems (das)* (pp. 66–70).

Ferreira, R., de Souza Cabral, L., Lins, R. D., Pereira e Silva, G., Freitas, F., Cavalcanti, G. D., et al. (2013). Assessing sentence scoring techniques for extractive text summarization.. *Expert Systems with Applications, 40*(14), 5755–5764.

Filippova, K. (2010). Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of the 23rd international conference on computational linguistics. COLING '10* (pp. 322–330). Stroudsburg, PA, USA: Association for Computational Linguistics.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the thirteenth international conference on machine learning* (pp. 148–156). San Francisco: Morgan Kaufmann.

Gambhir, M., & Gupta, V. (2016). Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 1–66.

Gibbons, J. D., & Chakraborti, S. (2011). *International encyclopedia of statistical science* (pp. 977–979)). Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.

Gillick, D., & Favre, B. (2009). A scalable global model for summarization. In *Proceedings of the workshop on integer linear programming for natural langauge processing. ILP '09* (pp. 10–18). Stroudsburg, PA, USA: Association for Computational Linguistics.

Glavaš, G., & Šnajder, J. (2014). Event graphs for information retrieval and multi–document summarization. *Expert Systems with Applications, 41*(15), 6904–6916.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning* (1st ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Greenbacker, C. F. (2011). Towards a framework for abstractive summarization of multimodal documents. In *Proceedings of the association for computational linguistics 2011 student session. HLT-SS '11* (pp. 75–80). Stroudsburg, PA, USA: Association for Computational Linguistics.

Haque, R., Naskar, S. K., Way, A., Costa-jussa, M. R., & Banchs, R. 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.

Haykin, S. (1998). *Neural networks: A comprehensive foundation* (2nd ed.). Upper Saddle River, NJ, USA: Prentice Hall PTR.

Hong, K., Conroy, J., Favre, B., Kulesza, A., Lin, H., & Nenkova, A. (2014). A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of the ninth international conference on language resources and evaluation (LREC'14)* (pp. 1608–1616). Reykjavik, Iceland: European Language Resources Association (ELRA).

Hong, K., Marcus, M., & Nenkova, A. (2015). System combination for multi-document summarization. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 107–117). Lisbon, Portugal: Association for Computational Linguistics.

John, G. H., & Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the eleventh conference on uncertainty in artificial intelligence* (pp. 338–345). San Mateo: Morgan Kaufmann.

Khan, A., Salim, N., & Kumar, Y. J. (2015). A framework for multi-document abstractive summarization based on semantic role labelling. *Applied Soft Computing, 30*, 737–747.

Le Cessie, S., & Van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied Statistics, 41*(1), 191–201.

Leite, D. S., & Rino, L. H. M. (2008). Combining multiple features for automatic text summarization through machine learning. In A. Teixeira, V. L. S. de Lima, L. C. de Oliveira, & P. Quaresma (Eds.), *Computational processing of the portuguese language. Lecture notes in computer science: Vol. 5190* (pp. 122–132). Springer-Verlag Berlin Heidelberg.

Li, C., Liu, Y., & Zhao, L. (2015). Using external resources and joint learning for bigram weighting in ILP-based multi-document summarization. . In R. Mihalcea, J. Y. Chai, & A. Sarkar (Eds.), *Proceedings of the north american chapter of the association for computational linguistics (NAACL)* (pp. 778–787). Association for Computational Linguistics.

Lin, C.-Y. (2004). Rouge: a package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop* (pp. 74–81). Barcelona, Spain: Association for Computational Linguistics.

Lins, R. D., Simske, S. J., de Souza Cabral, L., de França Silva, G., Lima, R., Mello, R. F., & Favaro, L. (2012). A multi-tool scheme for summarizing textual documents. In *Proceedings of the 11st IADIS international conference www/internet 2012* (pp. 1–8).

Liu, F., Flanigan, J., Thomson, S., Sadeh, N. M., & Smith, N. A. (2015). Toward abstractive summarization using semantic representations. In *The 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1077–1086). Denver, Colorado, USA

Lloret, E., & Palomar, M. (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.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: System demonstrations* (pp. 55–60).

Marujo, L., Ling, W., Ribeiro, R., Gershman, A., Carbonell, J., de Matos, D. M., et al. (2015). Exploring events and distributed representations of text in multi-document summarization. *Knowledge-Based Systems, 94*, 33–42.

Mausam, Schmitz, M., Bart, R., Soderland, S., & Etzioni, O. (2012). Open language learning for information extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. EMNLP-CoNLL '12* (pp. 523–534). Stroudsburg, PA, USA: Association for Computational Linguistics.

Meena, Y., Deolia, P., & Gopalani, D. (2015). Optimal features set for extractive automatic text summarization. In *Proceedings of the fifth international conference on advanced computing communication technologies (ACCT)* (pp. 35–40).

Meena, Y. K., & Gopalani, D. (2014). Analysis of sentence scoring methods for extractive automatic text summarization. In *Proceedings of the 2014 international conference on information and communication technology for competitive strategies. ICTCS '14* (pp. 51–56). New York, NY, USA: ACM.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into texts. In *Proceedings of conference on empirical methods on natural language processing (EMNLP)* (pp. 404–411). Barcelona, Spain: Association for Computational Linguistics.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM, 38*(11), 39–41.

Nenkova, A., & McKeown, K. (2012). A survey of text summarization techniques. . In C. C. Aggarwal, & C. Zhai (Eds.), *Mining text data* (pp. 43–76). Springer.

Nenkova, A., Vanderwende, L., & McKeown, K. (2006). A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval. SIGIR '06* (pp. 573–580). New York, NY, USA: ACM.

Neto, J. L., Freitas, A. A., & Kaestner, C. A. A. (2002). Automatic text summarization using a machine learning approach. In G. Bittencourt, & G. Ramalho (Eds.), *Advances in artificial intelligence. Lecture notes in computer science: Vol. 2507* (pp. 205–215). Springer-Verlag Berlin Heidelberg.

Ouyang, Y., Li, W., Lu, Q., & Zhang, R. (2010). A study on position information in document summarization. In *Proceedings of the 23rd international conference on computational linguistics: Posters. COLING '10* (pp. 919–927). Stroudsburg, PA, USA: Association for Computational Linguistics.

Owczarzak, K., Conroy, J. M., Dang, H. T., & Nenkova, A. (2012). An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of workshop on evaluation metrics and system comparison for automatic summarization* (pp. 1–9). Stroudsburg, PA, USA: Association for Computational Linguistics.

Palshikar, G. K., Deshpande, S., & Athiappan, G. (2012). Combining summaries using unsupervised rank aggregation. In A. F. Gelbukh (Ed.), *Proceedings of the conference on intelligent text processing and computational linguistics (cicling)*. In *Lecture notes in computer science: 7182* (pp. 378–389). Springer.

Platt, J. (1998). Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods - support vector learning*. MIT Press.

Quinlan, J. R. (1992). Learning with continuous classes. In *Proceedings of the australian joint conference on artificial intelligence* (pp. 343–348). World Scientific.

Quinlan, R. (1993). *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann Publishers.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on artificial intelligence. IJCAI'95* (pp. 448–453). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Saggion, H., & Poibeau, T. (2013). Automatic text summarization: past, present and future. In T. Poibeau, H. Saggion, J. Piskorski, & R. Yangarber (Eds.), *Multi-source, multilingual information extraction and summarization*. In *Theory and applications of natural language processing* (pp. 3–21). Springer-Verlag Berlin Heidelberg.

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika, 52*(3/4), 591–611.

Silva, G., Ferreira, R., Lins, R. D., Cabral, L., Oliveira, H., Simske, S. J., & Riss, M. (2015). Automatic text document summarization based on machine learning. In *Proceedings of the 2015 ACM symposium on document engineering. DocEng '15* (pp. 191–194). New York, NY, USA: ACM.

Simske, S. J. (2013). *Meta-Algorithmics: Patterns for robust, low cost, high quality systems* ((1st ed.). Wiley-IEEE Press.

Sipos, R., Shivaswamy, P., & Joachims, T. (2012). Large-margin learning of submodular summarization models. In *Proceedings of the 13th conference of the european chapter of the association for computational linguistics. EACL '12* (pp. 224–233). Stroudsburg, PA, USA: Association for Computational Linguistics.

Toffler, A. (1970). *Future shock*. Random House.

Torres-Moreno, J.-M. (2014). *Automatic text summarization*. John Wiley & Sons, Inc.

Wan, X., & Yang, J. (2008). Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval. SIGIR '08* (pp. 299–306). New York, NY, USA: ACM.

Zajic, D., Dorr, B. J., Lin, J. J., & Schwartz, R. M. (2007). Multi-candidate reduction: sentence compression as a tool for document summarization tasks. *Information Processing & Management, 43*(6), 1549–1570.