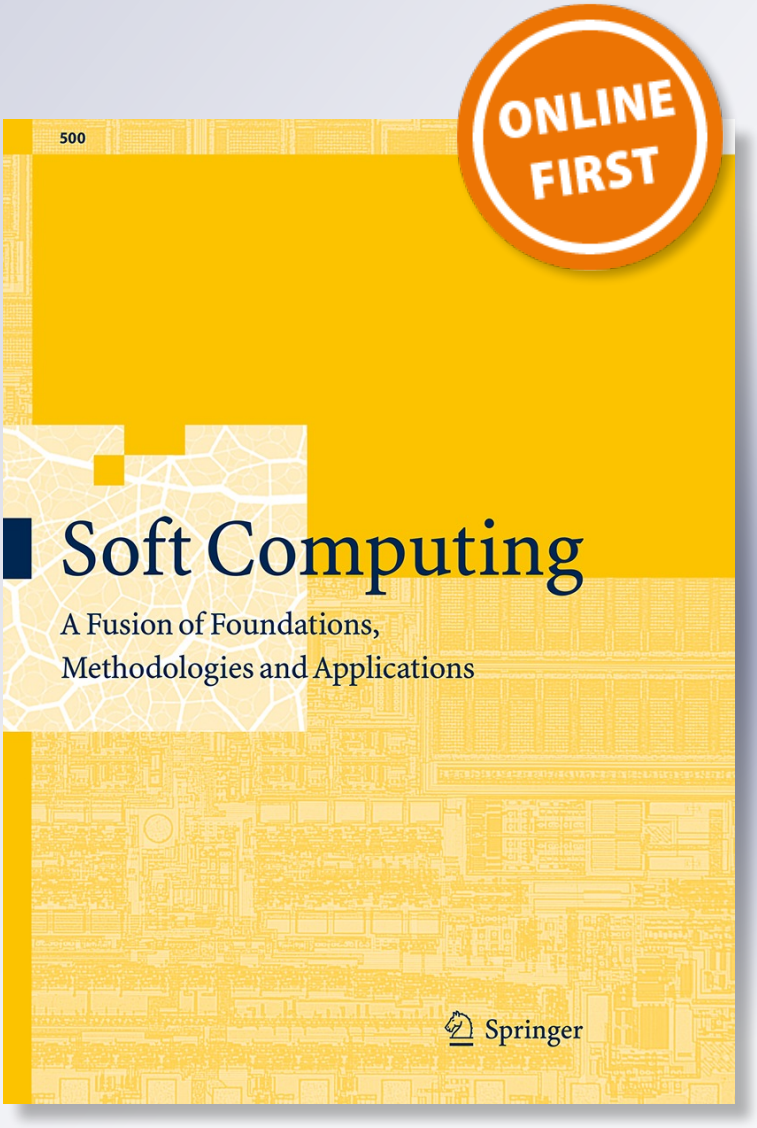
Exploring actor–object relationships for query-focused multi-document summarization

Mohammadreza Valizadeh & Pavel Brazdil

Soft Computing
A Fusion of Foundations, Methodologies and Applications

ISSN 1432-7643

Soft Comput
DOI 10.1007/s00500-014-1471-x



Springer

Soft Comput
DOI 10.1007/s00500-014-1471-x

METHODOLOGIES AND APPLICATION

Exploring actor–object relationships for query-focused multi-document summarization

Mohammadreza Valizadeh · Pavel Brazdil

Abstract Most research on multi-document summarization explores methods that generate summaries based on queries regardless of the users' preferences. We note that, different users can generate somewhat different summaries on the basis of the same source data and query. This paper presents our study on how to exploit the information regards how users summarized their texts. Models of different users can be used either separately, or in an ensemble-like fashion. Machine learning methods are explored in the construction of the individual models. However, we explore yet another hypothesis. We believe that the sentences selected into the summary should be coherent and supplement each other in their meaning. One method to model this relationship between sentences is by detecting actor–object relationship (AOR). The sentences that satisfy this relationship have their importance value enhanced. This paper combines ensemble summarizing system and AOR to generate summaries. We have evaluated this method on DUC 2006 and DUC 2007 using ROUGE measure. Experimental results show the supervised method that exploits the ensemble summarizing system combined with AOR outperforms previous models when considering performance in query-based multi-document summarization tasks.

Keyword User-based summarization · Actor–object relationship · Multi-document summarization · Ensemble summarizing system · Training data construction

Communicated by V. Loia.

M. Valizadeh (✉) · P. Brazdil
LIAAD INESC Tec, University of Porto, Porto, Portugal
e-mail: valizadehmr@gmail.com

P. Brazdil
FEP, University of Porto, Porto, Portugal
e-mail: pbrazdil@inescporto.pt

Published online: 14 October 2014

1 Introduction

Document summarization generates a short text for a single document or multiple documents. This summary should be informative and non-redundant and the process should be efficient. It means that summary should capture the important concepts of the original documents. Abstractive and extractive methods are two main approaches to summarize documents automatically. Abstractive methods are based on language processing and reformulating the original sentences while extractive methods concatenate the relevant sentences into the summary.

The relevant sentences are identified by ranking the sentences based on certain features. Therefore, the features and the ranking algorithms using those features are the core of extractive methods.

Recent research has exploited various features (e.g., similarity between two sentences, sentence length, sentence position, etc.) that are based on the general information of the data set regardless of users. It is possible to learn models of particular users and combine them into an ensemble summarizing system to generate the summary. This method (i.e., ensemble summarizing system) requires new features that enable to capture the users' preferences. Besides, the learning methods employed should be fast and efficient.

Graph-based models are widely used in extractive multi-document summarization systems that represent an unsupervised approach. Document set is represented as a graph in which the sentences are represented by nodes and the similarities between sentences are represented as edges. This paper introduces some features based on the graph topology that have not been used in the past research. These features permit to derive a model of how a particular user selects a sentence for a given summary. Here, we are referring to a supervised approach. We believe that the nodes correspond-

ing to important sentences selected by a user may be different for different users and hence the user-based approach is relevant.

The training data required can be constructed automatically on the basis of human summaries provided by each user. We note that in DUC 2006 and DUC 2007, there were 10 assessors/users. The user summaries and their corresponding document sets can be used to indicate the sentence importance scores in the source text, following Ouyang et al. (2011) who used ROUGE-1 Lin (2004). The training data and the introduced features are then used to learn the users' models (models of the assessors A-J) for each document set separately. In this paper, we have employed the feed-forward neural networks (NNs) in the learning process, as NNs are well-known and represent a quite successful learning method. Each user is characterized by several models, one per document set summarized by himself.

In addition, this research exploits that the hypothesis the actor–object relationship between sentences is useful when generating the summary. Our aim is to model this and select the sentences into the summary that have some coherence. We focus our attention to cases when some *object* appears in one sentence and more information about it is provided in another sentence. In the latter sentence the term typically has the role of an *actor* (i.e., subject or agent). Whenever such case is detected, the importance of the latter sentence is enhanced. Therefore, these sentences will have more chance to be selected for the summary. We describe this in detail further on.

We have carried out a series of experiments to evaluate the performance of the proposed approach. The results show that ensemble summarizing system that is combined with explicit identification of AOR outperforms the state-of-the-art systems for query-based multi-document summarization.

The rest of the paper is organized as follows. Section 2 discusses the previous work. In Sect. 3, the proposed method is described in detail. The experiments and evaluations are presented in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Related work

Most of the existing methods for multi-document summarization are extractive. As we know, sentence ranking algorithms play the main role in these methods. There are many sentence ranking algorithms that are described in previous studies that are based on graph-based methods and involve feature selection, machine learning techniques, clustering, etc. (Wan et al. 2006; Patil 2007; Ouyang et al. 2011, etc.).

In addition, various people have used machine learning techniques in document summarization (Chuang and Yang 2000; Mani and Bloedorn 1998; Neto et al. 2002). One of the first trainable summarizer systems was proposed by Kupiec et al. (1995) that used naïve Bayes classifier. Aone et al. (1999) also used naïve Bayes classifier, term frequency (tf) and inverse document frequency (idf) features in their system DimSum.

Some researchers explored decision trees as the learning method. Lin (1999) used decision trees and examined various features and studied their effect on sentence extraction and the summary. They used TIPSTER[1] data collection and exploited some features such as *query signature* (i.e., normalized score given to sentences depending on the number of query words that they contain), *IR signature* (the *m* most salient words in the corpus), *numerical data* (Boolean value 1 given to sentences that contained a number in them) and *proper name* (Boolean value 1 given to sentences that contained a proper name in them).

Conroy and O'leary (2001) exploited hidden Markov models to extract the important sentences for document summarization. They used *position of the sentence*, *number of terms* in the sentence and the *probability of sentence terms* based on the document terms.

Osborne (2002) showed that existing methods assume that the features are independent. He used log-linear methods and his system generated better summaries.

Svore et al. (2007) proposed a summarization system that exploited a neural network. They used a data set with 1,365 documents collected from CNN.com and ROUGE-1 to score the similarity between the human written sentence and a sentence in the summary. They used some features based on query logs from Microsoft news and Wikipedia.

Learning-to-rank models have recently been considered. Amini et al. (2005) proposed a learning-to-rank model for query-based single document summarization. Toutanova et al. (2007) introduced PYTHY system using more than 20 features and its results on DUC 2007 were very good. Ouyang et al. (2011) applied regression model to query-based multi-document summarization. Their system exploits a set of predefined features and estimates the importance of a sentence in a document set by support vector regression. The training data are constructed automatically from DUC collections and used to generate the models.

One of the most important requirements of the learning-to-rank approaches is sufficient training data (Ouyang et al. 2011). Generating this data is expensive and time consuming. As we know, there are several data collections with included summaries and they have been produced for automatic evaluations of the participating system in the competitions (e.g., DUC 2006, DUC 2007, etc.). The human summaries have been used by some researchers (Chuang and Yang 2000; Fisher and Roark 2006; Toutanova et al. 2007; Ouyang et al. 2011) to generate the training data to learn the ranking models. Based on these collections, some researchers master-

[1] See http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/.

minded a way to generate the training data with less overload (i.e., semi-automatic strategy).

The researchers concentrated on summarizing the data set based on the query regardless of the user that might use the summary. In other words, they generate a single model for the summaries of data set and the query. However, it is possible to generate different models for different users from the data set and the query. These can then be combined into a single model that is referred to as the *ensemble summarizing system*. The disadvantage of this method is that we need to have training data for each user.

Some researchers have explored the fact that the users generate specific log files, or can indicate a set of papers of interest (Diaz and Gervás 2007; Park and Cha 2008). However, this research is based on the user's interest and is not intended to identify the user's method of summarizing a document.

Moro and Bielikova (2012) proposed a personalized text summarization based on identification of important terms. They used comments added by readers as one of the sources of personalization. Yang et al. (2012) proposed a personalized automatic text summarization system for mobile learning. Their system helps mobile learners to retrieve and process information more quickly. They used probabilistic language modeling techniques to model a user. Xu et al. (2009) proposed user-oriented document summarization through vision-based eye tracking. They used the time that a user spent on a single word to read as an important feature in the learning phase. They used this information to predict the users' attention on the words in the documents.

This paper uses the automatic method for generating the training data set and then exploits various features based on the graph topology. The training data are used to acquire the user's model of document summarization and ensemble summarizing system with the recourse to machine learning methods. The experiments show that the ensemble system combined with AOR outperforms the state-of-the-art systems.

In the next section, the method is explained in more details.

3 The proposed ensemble method

Our ensemble summarization method generates several models for each user and combines them to generate a unique model. Therefore, training data for each user are needed to be able to apply machine learning techniques. Figure 1 shows the architecture employed here.

The proposed system (Fig. 1a) uses human summaries to generate the training data to learn the ranking models. Function *Generating features* computes values of the features of all sentences in the document set. Function *Extracting scores* computes a score (i.e., target value) of each sentence. More details are presented further on in this section (i.e., Training data construction). The tuple $<S_{1ij}, f_{11ij}, \ldots, f_{n1ij}, score_{1ij}>$ denotes a structure that includes the features and the related score (i.e., score is the target feature in the learning process) for sentence S_{1ij}. S_{kij} denotes sentence k of document D_{ij} and D_{ij} denotes document i of document set j. The results of these models (ranked lists of sentences) are then combined into a single ranked list that enables to generate the summary (Fig. 1b).

There are several data collections accompanied by human summaries which have been used for automatic evaluation of the participating systems in various competitions (e.g., DUC 2006, DUC 2007, etc.). It means that the users' models can be used to generate summary for a new document set (source text). The scheme used is illustrated in Fig. 1b. If the user has summarized m document sets, our system learns models from all document sets and uses these models (all except the models relative to the test document set), to generate the summary. Function *Merging* combines all generated ranked lists to obtain the combined ranked list.

The following subsections describe the important concepts and stages of the proposed method.

3.1 Feature extraction

Features play an important role in the process of generating the models. We need features that capture the user's preferences when selecting sentences. Our conjecture is that some of these should be related to the graph-based representation. After some considerations, we have selected 10 features to characterize the process of users' selection from the source data. They are described in the following.

3.1.1 Sentence length

Neither very short nor very long sentences may not be important. A very short sentence is normally not informative and a very long sentence wastes the resource of fixed summary length. Therefore, we use the sentence length as one of our features:

$$\text{sentence\_length}(S) = |\text{words}(S)| \quad (1)$$

where |words(S)| represents the number of the words in S.

3.1.2 Sentence length without stop-words

This feature shows how many informative words there are in the sentence. If the sentence has many stop-words, it will normally be less informative. Thus, we define this feature as follows:

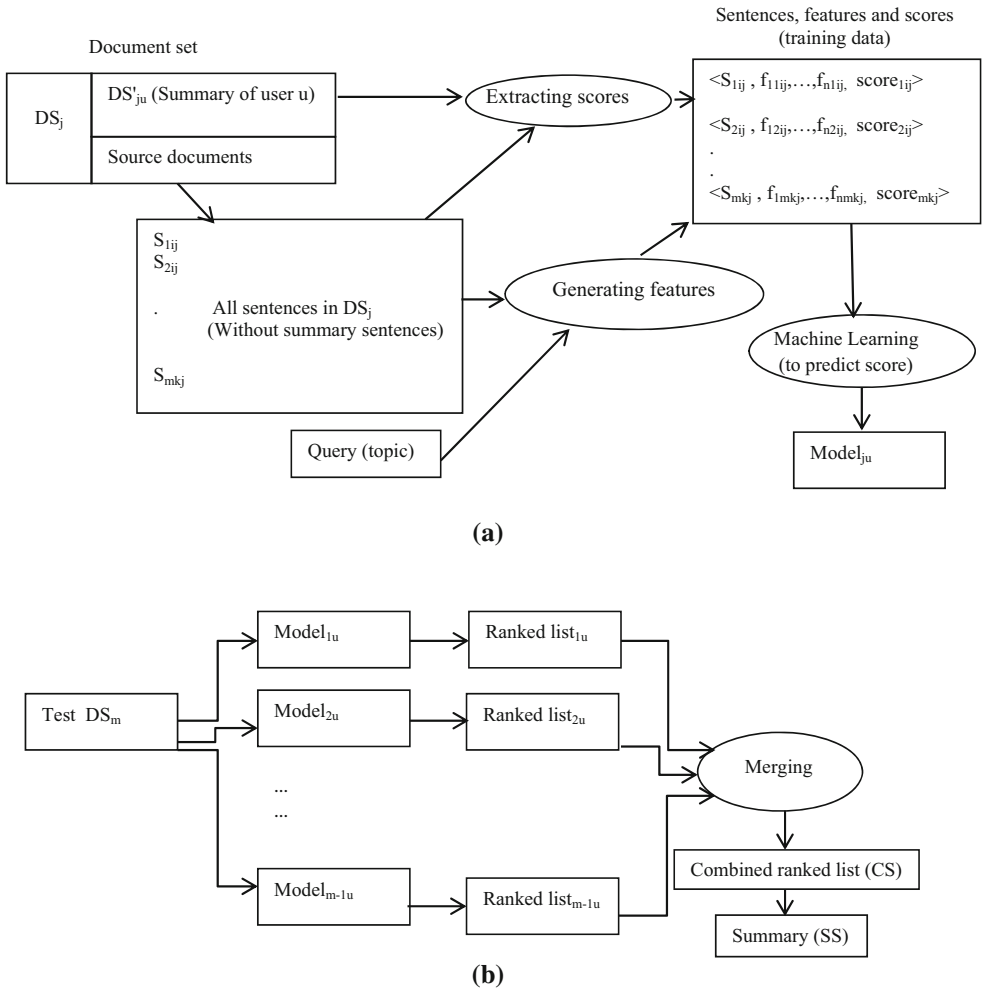


Fig. 1 The architecture of ensemble summarizing system for a document set. **a** Generating Model $_j u$ for data set DS$_j$ and the corresponding summary DS$'_{ju}$ created by user u. **b** Feeding test document set to users models and generating a ranked list

$$\text{length\_without\_stop} - \text{word(S)} = \text{sentence\_length(S)} - |\text{stopword(S)}| \quad (2)$$

where |stopword(S)| is the number of the stop-words in S.

3.1.3 Sentence radius

If the sentence is closer to the centroid of the corresponding document, it carries more information from the document (Valizadeh and Brazdil 2013, 2015). The radius is the distance between the sentence and the document centroid. The equation below provides the definition of this feature:

$$\text{radius(S)} = \left(\frac{\sum_{k=1}^{N} (\overrightarrow{X_k} - \overrightarrow{X_0})^2}{N} \right)^{\frac{1}{2}} \quad (3)$$

where N is number of sentences, vector $\overrightarrow{X_0}$ is the centroid and vector $\overrightarrow{X_k}$ represents a sentence in the vector-space model. The centroid is defined as:

$$\overrightarrow{X_0} = \frac{\sum_{k=1}^{N} \overrightarrow{X_k}}{N} \quad (4)$$

3.1.4 Average of TF-IDF

The well-known representation TF-IDF permits to highlight the importance of words in each sentence. We use the average of the TF-IDF weights of the words of the given sentence as a feature. Since sentences have different lengths, it would not make sense to sum TF-IDF of all words, because longer sentences would get a higher weight compared to shorter ones. Therefore, we have used the average value. The following equation shows how it is calculated:

$$\text{tfidf(S)} = \left(\sum_{wi} \text{tfidf}(w_i) \right) / n \quad (5)$$

where tfidf(w_i) is the tf-idf weight of the word w_i in the collection of documents and n is number of words in sentence S.

3.1.5 Sentence to query similarity

The summarizer is required to generate a summary to satisfy the user, while taking into account the given query (Q). Therefore, this feature is important and is defined as follows:

$$\text{sim\_query(S)} = \text{COS(S, Q)} \tag{6}$$

where COS(S,Q) computes the cosine similarity between the sentence (S) and the query (Q) (i.e., each document set has a *topic* (i.e., short description) that is called Query).

3.1.6 Sentence position

Normally authors introduce their main idea at the beginning of their texts. This observation can be exploited. Therefore, this feature is defined as follows (Patil 2007; Ouyang et al. 2011):

$$\text{position(S)} = 1 - \frac{i-1}{n} \tag{7}$$

where n is the total number of the sentences and S is the ith sentence in the document.

3.1.7 Sum of similarities between current sentence and other sentences

For computing this feature, we need the similarity between each sentence and all the other sentences in the given document set with weight (i.e., similarity between two sentences) greater than 0.05 to avoid the problem of a link by-chance. This feature covers some part of the graph topology and shows how much the current sentence is similar to the other sentences in the graph (i.e., as we know, the graph of the sentences is not directed). The following equation shows how this feature is computed:

$$\text{sim\_to\_others}(S) = \sum_{S_j \text{ data set}} \text{COS}(S, S_j) \tag{8}$$

where S in the current sentence, S_j is another sentence of the document set and COS computes the cosine similarity.

3.1.8 Sum of similarities between the current sentence and the top 5 sentences

This feature calculates the sum of the top 5 similarities between the current sentence and the other sentences. To compute this feature, we need the measure of similarity between each sentence and all the other sentences in the document set. This feature covers some part of the graph topology and shows how strong the sentence relationships are. We define this feature as follows:

$$\text{sim\_to\_top5}(S) = \sum_{S_j \text{ Top5}} \text{COS}(S, S_j) \tag{9}$$

where S in the current sentence, S_j is another data set sentence and COS computes the cosine similarity.

3.1.9 Number of nonzero links

When a sentence has many links to the other sentences, it means that this sentence is similar to many other sentences and, therefore, it is important. Important sentences are often re-phrased and repeated by the authors of texts.

This feature shows how many links there are between this sentence and the other sentences in the document set with weight (i.e., similarity between the two sentences) greater than 0.05. Our assumption is that the more links there are (the more relationships it has) the more important this sentence is. We define this feature as follows:

$$\text{nonzero(S)} = \left|\text{Link}(S, S_j) >= 0.05\right| \; S_j \text{ is member of the graph nodes} \tag{10}$$

The equation shows the number of outgoing links from S with more than 0.05 weight.

3.1.10 Sentence rank of T-LexRank

The intuition here is that if a sentence get a high rank by T-LexRank (Otterbacher et al. 2005), it may be more useful for the final summary. Non-supervised summarization methods, such as T-LexRank, use just this feature. We see no reason why this feature should not be reused in a supervised setting. This feature is defines as follows:

$$\text{rank\_by\_TLexRank(S)} \quad \text{represents the rank of } S. \tag{11}$$

Let us compare the features used in our system and some other systems. Ouyang's system used seven features including three query-dependent ones (i.e., *word matching*, *semantic matching*, *named entity matching*) and four query-independent features (i.e., *TF-IDF, named entity, stop-word penalty, sentence position*). This indicates that Ouyang et al. focused on the relationship between the query and the documents and their features do not cover the users' behavior during the process of sentence selection. However, our features cover this. Table 1 shows the comparison between our features and some other researchers. Some features are not exactly similar, but compute the same entity in different way

Table 1 Used features in different systems

Feature	Ouyang's system	Toutanova's system	Our system
Sentence length		*	*
Sentence length without stop-words	*		*
Sentence position	*	*	*
Sentence Radius			*
Average of TF-IDF	*	*	*
Sentence to query similarity	*	*	*
Sum of similarities between current sentence and other sentences			*
Sum of similarities between the current sentence and the top 5sentences			*
Number of nonzero links			*
Sentence rank of T-LexRank			*

(e.g., stop-word penalty and sentence length without stop-words).

Toutanova's system uses more features than the ones showed here, but these are based on syntax and we did not show them here. This table shows that we focus on the features that are extracted from the graph. When a specific user selects some sentences to generate the summary, these features capture the relationships between these sentences, represented as nodes, and other sentences in the graph.

3.2 Training data construction

The training data supplied to the learning systems should include the sentences together with their importance score representing the target value. The importance score should estimate how strongly the source sentence is related to the sentences in the human-supplied summary. In this paper, we reuse the strategy of Ouyang et al. (2011) that was used for query-based summarization. The idea is that if human summaries are acceptable, the sentences in the documents that are more similar to them should be acceptable as well. The higher the similarity, the higher the score attributed to the sentence. Our system searches for the sentence in the summary that has the highest similarity to a given sentence S in the source text and this score is attributed to S.

Ouyang et al. (2011) used this idea to construct the training data and generate a model for a given data set. They did not take into account the person who carried out the summarization (the summarizer). If we consider DUC 2006 and DUC 2007, they have 50 and 45 document sets, respectively, and each document set is accompanied by 4 human summaries. In addition, there are 10 summarizers for each DUC. Consequently, each summarizer has summarized 20 document sets of DUC 2006 and 18 document sets of DUC 2007, respectively (i.e., each summarizer has 20 sample document sets of DUC 2006 and 18 sample document sets of DUC 2007). Therefore, there is enough information to examine the hypothesis that different people expect to obtain a different summary for the same document set and query.

The data obtained from each summarizer (i.e., summaries) are explored separately. It means that each sentence S_{kij} in document set DS_j is assigned an importance score which is computed by ROUGE-1 on the basis of the related summary $\mathrm{DS}'_{\mathrm{ju}}$ done by user u.

Therefore, four models are generated for each document set because each document set of DUC 2006 and DUC 2007 has four different human summaries (see Fig. 1a).

One issue is whether the quantity of the training data is sufficient to obtain good models. Considering that each summarizer has summarized about 20 document sets, the volume seems sufficient to construct a useful model. The results confirm this.

Another interesting issue is whether the input of a particular summarizer is trustworthy and useful. We have indeed used the DUC-2006 and DUC-2007 data without questioning its quality. However, the trustworthiness /usefulness could be assessed. It would be possible to evaluate how a given model (i.e., of a particular summarizer) performs on the data of other summarizers. If the results were sub-standard in comparison with other models, the model could be simply dropped. This study could be carried out as a part of future work.

3.3 Model learning techniques

There are many machine learning techniques that have been used in the past research and each one has some advantages and disadvantages. Some of them were mentioned in the section discussing related work. We have experimented with several different ones, but in this paper we report the results with

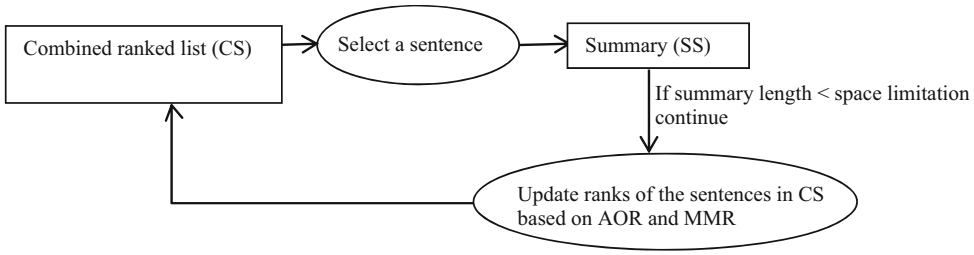


Fig. 2 Generating the summary (SS) from combined ranked list (CS)

back-propagation neural network that was used for learning the users' models.

3.4 Feeding a user's models and combining the results

Figure 1a shows the architecture of the system that generates different regression models (i.e., user-based) for different document sets and different users. Let us now see how these models can be used to generate a summary for a new text (e.g., test document or document set) (Fig. 1b). The test document is fed to the all users' models and each model generates a ranked list. Our aim is to combine these (by merging) to generate a single ranked list. This can be done in different ways. One simple way is by adding the scores of each sentence in the different ranked lists to generate a unique score for each sentence. This is what we have done (see Fig. 1).

It is necessary to explain why we have generated several models for each user (i.e., summarizer) instead of just a single model. As we know, each document set may include some specific issues which are summarized in a specific way. Therefore, if we combine sentences of all document sets and all their summaries into a single summary and generate a model for this, we may lose some information. This is obviously a conjecture.

The experimental results described in the next section provide an evidence that this conjecture is presumably correct.

3.5 Generating the summary with updating the ranked list

The summary (SS) is generated sentence by sentence from the combined ranked list (CS) for the test document set. Figure 2 illustrates this process. The sentence with the highest score is selected for the summary (SS). After this, the combined ranked list (CS) is updated taking into account the sentence chosen for SS, AOR and MMR. Here, MMR represents the Maximum Marginal Relevance approach described by others (Carbonell and Goldstein 1998).

This process is continued until the summary space limitation has been reached (e.g., 250 words for DUC 2006 and DUC 2007).

Updating the combined ranked list CS includes two steps. The first one involves updating the list based on AOR. The aim is to generate a summary that would enhance coherence, but not redundancy. One method to model this relationship between sentences is by detecting actor–object relationship (AOR). The sentences that satisfy this relationship have their importance value enhanced. This approach requires that all words/terms be annotated with tags. We have used the Stanford dependency parser to do this. Therefore, we explain some aspects of the parser that are important to carry out this task. The parser was produced by the Stanford Natural Language Processing Group (De Marneffe et al. 2006). It is a natural language parser that analyzes the grammatical structure of sentences. This parser uses knowledge of language acquired from hand-parsed sentences to produce the most likely analysis of new sentences. It can identify phrases, an actor or object of a verb etc.. We describe some concepts below which are important to describe the solution adopted. First, we show some grammatical tags that are relevant here. They are shown in Table 2.

Table 2 Extracted tags by Stanford parser

Tags in Stanford dependency parser	Description
Objects	
dobj	Direct object: The direct object of the verb
iobj	Indirect object: The indirect object of of the verb
Actors	
nsubj	Nominal actor: A nominal actor is a noun phrase which is the syntactic actor of a clause
nsubjpass	Passive nominal actor: A noun phrase which is the syntactic actor of a passive clause
agent	Agent: The complement of a passive verb introduced by the preposition "by" who executes the action

In Table 2, *dobj* and *iobj* represent objects of sentences and *nsubj*, *nsubjpass* and *agent* represent the *actor* or *subject* of sentences. These two categories are attached to the corresponding sentences. So, a particular sentence may be accompanied by a list of assertions. One of these may be, e.g., dobj(seize-47, compound-51) (see Fig. 3). This assertion means that the word "*seize*" has "*compound*" as a direct object.

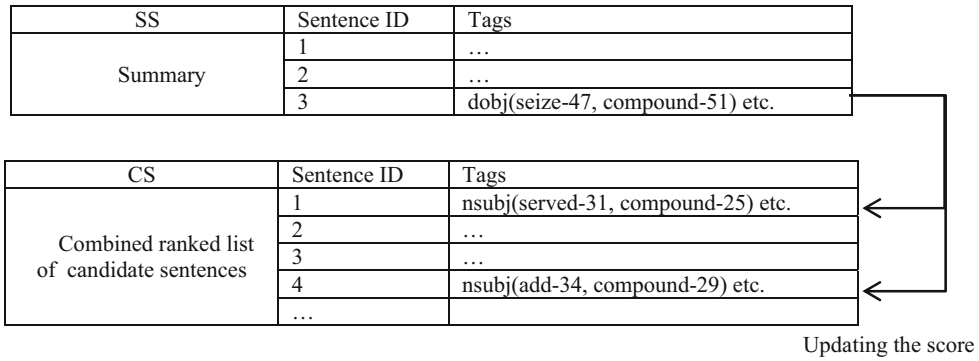


Fig. 3 Detecting certain dependency parser patterns

What interests us to detect patterns of the following kind:

$$\begin{aligned}&\text{dobj}(t_i - p_i, \mathbf{t}_\text{j} - p_j) \in \text{DP}(S_i|S_i \in \text{SS})\\ &\quad \Lambda\text{nsubj}(t_k - p_k, \mathbf{t}_\text{j} - p_l) \in \text{DP}(S_j|S_j \in \text{CS})\end{aligned} \tag{12}$$

where $t_i - p_i$ represents some term t_i at position p_i and $DP(S_i|S_i \in SS)$ represents the result of applying the dependency parser to sentence S_i in the summary. We note that the pattern (12) includes the same term t_j in two different positions. One instance if this pattern representing the situation in Fig. 3 is:

$$\begin{aligned}&\text{dobj}(\text{seize} - 47, \mathbf{compound} - 51) \in \text{DP}(S_i|S_i \in \text{SS})\\ &\quad \Lambda\text{nsubj}(\text{served} - 31, \mathbf{compound} - 25) \in \text{DP}(S_j|S_j \in \text{CS}))\end{aligned} \tag{13}$$

If a sentence is identified satisfying the pattern shown, the combined ranked list is updated. The corresponding score of the sentence in the combined ranked list is updated using equation (14). The increased score will increase the chance that this sentence will be selected into the summary. The score is modified as follows:

$$\text{Score}(S_i) = \text{Score}(S_i) + \theta(\text{Score}(S_{\text{highscore}}) - \text{Score}(S_i)) \tag{14}$$

where, $\text{Score}(S_i)$ denotes the score of sentence S_i in the ranked list CS, $\text{Score}(S_{\text{highscore}})$ denotes the score of the selected sentence for the summary in the previous round and θ is a parameter which determines the influence of this rule. Setting it to a high value means that the user would like AOR to have a strong effect on the sentence selection. The right value of this parameter needs to be chosen. This is described in Sect. 4 where we discuss the experiments.

Figure 3 shows that the object of selected sentence (i.e., sentence number 3 in the summary) is the nominal subject of the sentences number 1 and 4 in the ranked list CS. Consequently, the scores of these sentences are updated by 14.

After updating the sentence scores, the highest scored sentence in the combined ranked list CS is selected to be included in the summary. This process is continued until the length limitation of the summary has been reached.

Figure 4 shows a summary generated with AOR. It illustrates the effects of using AOR. The rank of some sentences in the summary has been altered. Symbol '=' shows that these sentences have equal score to the sentences in summary without AOR, symbol '+' indicates that these sentences have been promoted from their positions or added to the summary. These are for instance sentences 3 and 4. The words to which pattern (12) applies are highlighted by underlining. Symbol '−' shows that scores of corresponding sentences have been reduced (for instance sentence 5).

As we can see, in sentence 3 the word "*leaders*" is the actor and this word is the object of sentence 2. This is captured by the tags:

Sentence 1: dobj(seize-47, compound-51), Sentence 2: dobj(ordered-39, leaders-40)

The second change in Summary with AOR is sentence number 4. This sentence has been added to the summary because word "*compound*" is subject of this sentence and it is object of sentence 1. In this case, the relevant tags are:

Sentence 3: nsubj(inspire-31, leaders-29), Sentence 4: nsubj(served-31, compound-25)

The strategy adopted helps to generate a coherent summary but, since the summary length is limited, redundant sentences could be introduced. These waste the summary resource length and hence the opportunities to include more relevant information can be missed. Therefore, to solve this problem, we adopt the MMR approach (Carbonell and Goldstein 1998) with a given threshold (e.g., 0.7, as in the past research). If the similarity between the sentence selected and any sentence already existing in the summary is less than the given threshold, the sentence is added to the summary, but otherwise the addition is blocked.

4 Experiments and results

Our proposed method—the ensemble summarizing system—combined with AOR has been applied for query-based multi-document summarization. The experiments have been car-

Summary with AOR:

= 1- COEUR d'ALENE, Idaho _ Morris S. Dees Jr., who has won a series of civil rights suits against the Ku Klux Klan and other racist groups in a campaign to put them out of business, came to court here Monday to try to seize the Aryan Nations compound that has nurtured white supremacists for more than 20 years.

= 2- COEUR d'ALENE, Idaho _ Issuing a verdict that civil rights organizations hope will bankrupt one of the nation's largest white-supremacist groups and limit its ability to preach hate, a state jury in northern Idaho Thursday ordered leaders of the Aryan Nations to pay more than $6 million to the victims of an attack two years ago by men who were serving as security guards at the group's compound near here.

+ 3- Since co-founding the Southern Poverty Law Center in 1971, Dees has wielded the civil lawsuit like a buck knife, carving financial assets out of hate group leaders who inspire followers to beat burn and kill.

+ 4- COEUR D'ALENE, Idaho (AP) -- A lawyer who specializes in bankrupting hate groups is going after the Aryan Nations, whose compound in the Idaho woods has served as a clubhouse for some of America's most violent racists.

- 5- The Southern Poverty Law Center represented the predominantly black Macedonia Baptist Church in Clarendon County, which won a $21.5 million judgment against the Christian Knights and Klan members after the church was burned four years ago.

- 6- In 1987, Dees won a $7 million verdict against a Ku Klux Klan organization over the slaying of a 19-year-old black man in Mobile, Ala., forcing the group to turn over its headquarters building.

Fig. 4 One example of a summary showing the effects of AOR

ried out on DUC 2006[2] and DUC 2007. All the documents and queries of the DUC datasets have been pre-processed by sentence segmentation and word splitting. Words have been stemmed by Porter Stemmer (Porter 1980) and stop-words have also been removed. The representation TF*IDF was used and cosine similarity measure was used to compute the similarity of pairs of sentences and the similarity of sentences–query pairs. To avoid the problem of a link by-chance (Wei et al. 2008, 2010) that happens when two sentences share one or two common words, we have set a small threshold, 0.05, and do not consider the links which have a lower value than this threshold.

We have used the proposed methods described in Sect. 3 and carried out the query-based summarization task which is limited to 250 words for DUC 2006 and DUC 2007.

The MATLAB implementations were used. The neural network[3] used had 2 hidden layers and 10 nodes per each layer. In addition, Stanford dependency parser[4] was used to identify the sentence tags.

We have carried out also experiments to show the importance of the proposed features.

Automatic evaluation toolkit (i.e., ROUGE-1.5.5) that is the state-of-the-art of automatic summarization evaluation based on N-gram comparison was used. ROUGE evaluates the summaries by comparing them with human summaries (e.g., it uses bigram matching for ROUGE-2 and skip-bigram-based matching with maximum skip distance for ROUGE-SU4). The ROUGE parameters were: -e -n 2 -x -a -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -d (Lin 2004). The tables presented further on show the results in the form of the average recall scores of ROUGE-2 and ROUGE-SU4, together with the 95 % confidential intervals in parentheses.

Figure 5 shows the results of experiments on parameter θ. The values of θ were varied (X axis) to see what would be its effect on ROUGE-2 or ROUGE-SU4 (Y axis). These experiments reveal that the best value of parameter θ is between 0.2 and 0.3. Therefore, we have set this parameter to 0.25.

Figure 5 shows that it is advantageous to use AOR, as the value of ROUGE indicates. Not using AOR is equivalent to theta = 0 and for this value the values of ROUGE-2 and ROUGE-SU4 are lower. It means that the hypothesis that AOR can improve the summary accuracy has been confirmed.

Table 3 shows the results of the proposed algorithm (ensemble system) with AOR and without AOR). In addition, the table provides a comparison with T-LexRank algorithm (i.e., considered as a benchmark), Ouyang's system (Ouyang et al. 2011) that can be considered as a state-of-the-art system and our previous Density-BasedQ system (Valizadeh and Brazdil 2013) that achieved good results in query-based multi-document summarization. Also, we have compared our system reported here to the results of 32 participating systems of DUC 2006 and the corresponding results are shown in Table 4.

Table 3 shows that ensemble system both with and without AOR outperforms the state-of-the-art systems and confirms that learning user's models and updating the combined ranked list based on AOR does improve performance. In addition, the results show that proposed systems outperform the other methods. The ensemble system with AOR shows 5.5 and 2.9 % improvement on ROUGE-2 and ROUGE-SU4, respectively, when compared to Ouyang's system. Since the reference summaries are written by humans, we can suppose that they have high coherence and readability. The results confirm that our method generates the summaries that are

[2] More details about DUC can be found at http://duc.nist.gov.

[3] Newff function in MATLAB.

[4] http://nlp.stanford.edu/software/lex-parser.shtml.

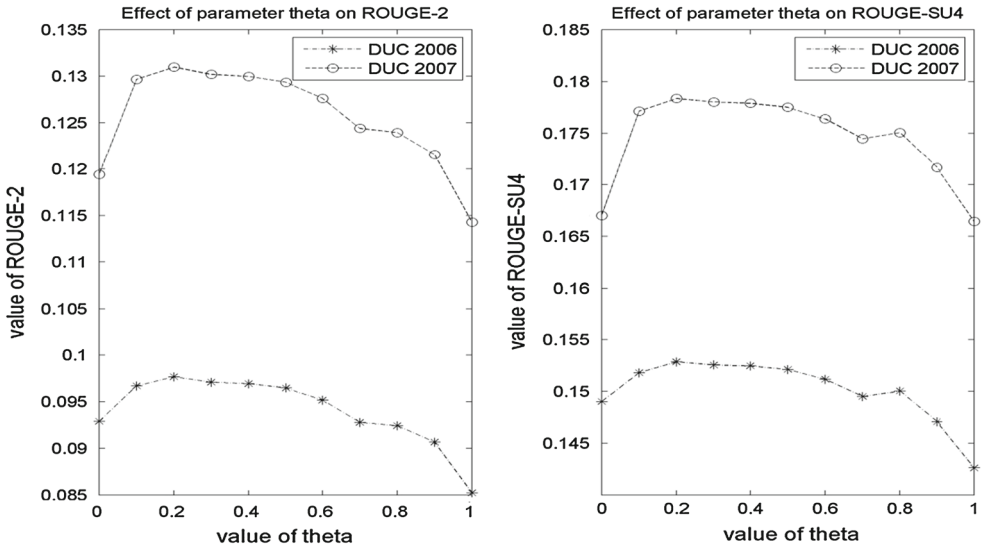


Fig. 5 Tuning parameter theta based on DUC 2006 and DUC 2007

Table 3 Model evaluation on DUC 2006 data set

	ROUGE-2	ROUGE-SU4
Ensemble system-AOR	**0.0977** (0.0935, 0.1023)	**0.1528** (0.1490, 0.1569)
Ensemble system	**0.0929** (0.0887, 0.0975)	**0.1490** (0.1452, 0.1531)
Ouyang's system	0.0926 (0.0883, 0.0969)	0.1485 (0.1443, 0.1525)
T-LexRank	0.0856 (0.0813, 0.0899)	0.1395 (0.1353, 0.1438)
Density-BasedQ	0.0907 (0.0867, 0.0947)	0.1444 (0.1404, 0.1476)

Bold values indicate the result of proposed system

Table 4 Comparison with systems participating in the DUC 2006

	ROUGE-2	ROUGE-SU4
Human	0.1001	0.1648
Ensemble system-AOR	**0.0977**	**0.1528**
S24	0.0950	0.1534
Ouyang's system	0.0926	0.1485
S15	0.0900	0.1448
S12	0.0892	0.1457
NIST Baseline	0.0491	0.0962

Bold values indicate the result of proposed system

more similar to the reference summaries and, therefore, the coherence of these summaries must have been improved.

Table 4 shows that our ensemble system-AOR outperforms the baseline system and also ranks 1st out of 32 based on ROUGE-2 and ROUGE-SU4 in DUC 2006.

We have performed also a series of experiments on DUC 2007 with the same setup as in the previous experiments. Tables 5 and 6 show the results.

Table 5—similarly as Table 3—shows that ensemble system with and without AOR outperforms the state-of-the-art systems and confirms that learning user's models and updating the combined ranked list based on AOR can improve performance. In addition, Table 5 results show that proposed ensemble system that uses AOR outperforms the other methods. Table 5 shows 12.1 and 5.7 % improvement on ROUGE-2 and ROUGE-SU4, respectively, compared to Ouyang's system.

Table 6 shows that our ensemble system-AOR outperforms the baseline system and also ranks $1st$ out of 32 based on ROUGE-2 and ROUGE-SU4 in DUC 2007.

For exploring the effect of learning the users' model we have carried out specific experiments. If the user (i.e., the summarizer) has summarized m document sets, our system learns $m - 1$ models and uses these models to generate the summary of the remaining document set in a leave-one-out style. We refer to this as user-based model. This method is applied to all document sets and all users. In addition, we evaluated the ensemble system in a leave-one-out style for users/summarizers. We have used models of $m - 1$ users evaluated the generated summary by the summary produced by summarizer m (leave-one-out ensemble system). The following tables show the results.

Tables 7 and 8 reveal that user-based summarizing system produces higher quality summaries when compared to the ensemble system because the learning of the models of users is done separately. The results again confirm that learning the users' models separately can capture more information

Table 5 Model evaluation on DUC 2007 data set

	ROUGE-2	ROUGE-SU4
Ensemble system-AOR	**0.1270** (0.1236, 0.1299)	**0.1746** (0.1713, 0.1782)
Ensemble system	**0.1194** (0.1161, 0.1231)	**0.1680** (0.1636, 0.1727)
Ouyang's system	0.1133 (0.1084, 0.1164)	0.1652 (0.1608, 0.1695)
T-LexRank	0.1051 (0.1007, 0.1094)	0.1560 (0.1516, 0.1599)
Density-BasedQ	0.1140 (0.1105, 0.1175)	0.1690 (0.1649, 0.1731)

Bold values indicate the result of proposed system

Table 6 Comparison with systems participating in DUC 2007

	ROUGE-2	ROUGE-SU4
Human	0.1289	0.1840
Ensemble system-AOR	**0.1270**	**0.1746**
S15	0.1239	0.1750
S29	0.1201	0.1694
S4	0.1181	0.1679
S24	0.1176	0.1743
Ouyang's system	0.1133	0.1652
S13	0.1115	0.1630
NIST Baseline	0.0599	0.1036

Bold values indicate the result of proposed system

than learning one model for all the users. Also, these tables show that ensemble system and leave-one-out version of it produce almost the same result. It means that eliminating one user does not have more effect on the ensemble system since it uses combination of many models.

These results show that the proposed idea is indeed useful for query-based multi-document summarization.

In addition, the results emphasize that learning the users' models separately can lead to better performance when compared to Ouyang's system that generates a single model for all users.

5 Conclusion

The main contribution of this paper is to introduce ensemble system combined with AOR. This system can be trained based on the user's way of summarizing. The results show that this system achieves a better performance than several recent systems described in literature. This system uses new features that are based on graph topology permitting to capture the user's behavior in the sentence selection process. The results on DUC 2006 and DUC 2007 support this claim. It is shown that the proposed system outperforms various state-of-the-art systems. These results confirm that if we consider the users and take them into account, this can improve the quality of the summaries significantly. In addition, AOR improves ROUGE values significantly. AOR enables to detect certain aspects of coherence among sentences, which we believe contributes positively towards higher quality of summaries.

The proposed system uses human summaries as the training data and generates a global model (i.e., ensemble-based system) and uses it for all new users. Therefore, it does not require that specific summaries be provided for each new user.

Comparing the proposed system with some recent systems reveals that our system has some main differences:

Table 7 Evaluation of DUC 2006 data set

	ROUGE-2	ROUGE-SU4
User-based	0.0933 (0.0894, 0.0972)	0.1492 (0.1453, 0.1531)
Ensemble system	0.0929 (0.0887, 0.0975)	0.1490 (0.1452, 0.1531)
Leave-one-out Ensemble system	0.0927 (0.0884, 0.0972)	0.1487 (0.1449, 0.1528)
Ouyang's system	0.0926 (0.0883, 0.0969)	0.1485 (0.1443, 0.1525)
T-LexRank	0.0856 (0.0813, 0.0899)	0.1395 (0.1353, 0.1438)

Table 8 Evaluation of DUC 2007 data set

	Average ROUGE-2	Average ROUGE-SU4
User-based	0.1219 (0.1181, 0.1257)	0.1695 (0.1650, 0.1739)
Ensemble system	0.1194 (0.1161, 0.1231)	0.1680 (0.1636, 0.1727)
Leave-one-out Ensemble system	0.1189 (0.1153, 0.1223)	0.1673 (0.1630, 0.1721)
Ouyang's system	0.1133 (0.1084, 0.1164)	0.1652 (0.1608, 0.1695)
T-LexRank	0.1051 (0.1007, 0.1094)	0.1560 (0.1516, 0.1599)

when the system generates separate models for each user and each document set, the models generated are better. This is because, when we generate a model for a given document set, many new intra-relationships between its sentences are created and these affect the inter-relationships. Furthermore, the summaries in the DUCs are related to the document set, not data set. Therefore, more accurate results are obtained if we compute the features values and sentence scores based on the appropriated document sets. In addition, our system improves accuracy and presumably coherence of the generated summary using AOR.

To our best knowledge no one has proposed a specific measure to evaluate the coherence. Our evaluation relies on ROUGE scores to reference summaries. Thus, we can only infer that the coherence of summaries has been improved. However, as the reference summaries have been written by humans, we can expect that they have high coherence and readability. When our generated summaries become more similar to these reference summaries, it means that their coherence has probably been improved too. However, it would be interesting to try to devise specific measures that can estimate coherence. This could be an object of future work.

We have used MMR for the last step of our method to select the sentences. It is foreseeable that if we used the improved version of MMR (e.g., Probabilistic Latent Maximal Marginal Relevance (Guo and Sanner 2010), the performance would be improved further. ROUGE-2 and -SU4 are the natural choice, although they do not capture how much the generated summary is semantically similar to the golden standard summary. We expect that some progress will be made on this in future, but in our view it exceeds the aims of this paper.

In the future work, we plan to examine further the effectiveness of exploiting other types of patterns resulting from using dependency parsers in a user-based summarization system and investigate their effectiveness.

Acknowledgments This work is funded (or part-funded) by the ERDF—European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT—Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project "FCOMP-01-0124-FEDER-022701"

References

Amini MR, Usunier N, Gallinari P (2005) Automatic text summarization based on word-clusters and ranking algorithms. In: Losada DE, Fernández-Luna JM (eds) ECIR 2005, LNCS, vol 3408. Springer, Heidelberg, pp 142–156

Aone C, Gorlinsky J, Larsen B, Okurowski ME (1999) A Trainable Summarizer with Knowledge Acquired from Robust NLP Techniques. In: Mani I, Maybury M (eds) Advances in Automatic Text Summarization. MIT Press, Cambridge, pp 71–80

Carbonell GJ, Goldstein J (1998) The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval, Melbourne Australia, pp 335–336

Chuang WT, Yang J (2000) Extracting sentence segments for text summarization: a machine learning approach. In: Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, pp 152–159

Conroy JM, O'leary DP (2001) Text summarization via Hidden Markov Models. In: Proceedings of SIGIR '01, New York, NY, USA, ACM SIGIR, pp 406–407

Diaz A, Gervás P (2007) User-model based personalized summarization. Inf Process Manag 43(6):1715–1734

Fisher S, Roark B (2006) Query-focused summarization by supervised sentence ranking and skewed word distributions. In: Document understanding conference. http://duc.nist.gov

Guo S, Sanner S (2010) Probabilistic latent maximal marginal relevance. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '10). ACM, New York, NY, USA, pp 833–834

Kupiec J, Pedersen J, Chen F (1995) A Trainable Document Summarizer. In: Proceedings of the 18th annual international conference ACM SIGIR, pp 68–73

Lin C-Y (1999) Training a selection function for extraction. In: Proceedings of the Eighteenth Annual International ACM Conference on Information and Knowledge Management (CIKM), Kansas City, Kansas, New York, NY, USA, ACM, pp 55–62

Lin C (2004) ROUGE: A Package for Automatic Evaluation of Summaries. In: Proceedings of workshop on text summarization, Branches Out, Post-conference workshop of ACL, Barcelona, Spain

Mani I, Bloedorn E (1998) Machine learning of generic and user-focused summarization. In: Proceedings of the 15th national/10th conference on artificial intelligence/innovative applications of artificial intelligence. AAAI Press, Madison, pp 820–826

De Marneffe M-C, MacCartney B, Manning C (2006) Generating Typed Dependency Parses from Phrase Structure Parses. In: Proceedings of Language Resources and Evaluation Conference

Neto JL, Freitas AA, Celso AA (2002) Kaestner. Automatic text summarization using a machine learning approach. In: Proceedings of the 16th Brazilian symposium on artificial intelligence: Advances in artificial intelligence. Springer-Verlag press, pp 205–215

Moro R, Bielikova M (2012) Personalized text summarization based on important terms identification. In: Proceeding of 23rd International conference of Database and Expert Systems Applications (DEXA), IEEE, Vienna, Austria, pp 131–135

Osborne M (2002) Using maximum entropy for sentence extraction. In: Proceedings of the ACL'02 Workshop on Automatic summarization, Morristown, NJ, USA, publisher: Association for Computing Linguistics, pp 1–8

Otterbacher J, Erkan G, Radev DR (2005) Using randomwalks for question-focused sentence retrieval. In: Proceedings of the human language technology conference/conference on empirical methods in natural language processing, publisher: Association for Computational Linguistics, pp 915–922

Ouyang Y, Li W, Li S, Lu Q (2014) Applying regression models to query-focused multi-document summarization. Inf Process Manag 47(2):227–237

Park S, Cha B (2008) Query Based Personalized Summarization Agent Using NMF and Relevance Feedback. In: Proceedings of the 2008 Third International Conference on Convergence and Hybrid Information Technology, vol 02 (ICCIT '08), IEEE Computer Society, Washington, DC, USA, pp 779–784

Patil K, Brazdil P (2007) SumGraph: Text Summarization using Centrality in the Pathfinder Network. Int J Comput Sci Inf Syst 2(1):18–32

Porter M (1980) An algorithm for suffix stripping. Progr Electron Libr Inf Syst 14(3):130–137

Svore K, Vanderwende L, Burges C (2007) Enhancing single-document summarization by combining RankNet and third-party sources. In: Proceedings of the EMNLP-CoNLL, Association for Computational Linguistics (ACL), pp 448–457

Toutanova K, Brockett C, Gamon M, Jagarlamudi J, Suzuki H, Vanderwende L (2007) The PYTHY summarization system: Microsoft research at DUC 2007. In: Document understanding conference 2007. http://duc.nist.gov

Valizadeh M, Brazdil P (2013) Density-Based Graph Model for Multi-Document Summarization. In: Proceedings of Portuguese Conference on Artificial Intelligence, EPIA2013, Azores, Portugal, pp 480–491

Valizadeh M, Brazdil P (2015) Density-Based Graph Model Summarization: Attaining better Performance and Efficiency, to appear in journal of Intelligent Data Analysis, IOS press

Wan X, Yang J, Xiao J (2006) Using Cross-Document Random Walks for Topic-Focused Multi-Document Summarization. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, IEEE, pp 1012–1018

Wei F, Li W, Lu Q, He Y (2010) A document-sensitive graph model for multi-document summarization. Knowl Inf Syst 22(2):245–259

Wei F, He Y, Li W, Lu Q (2008) A Query-Sensitive Graph-Based Sentence Ranking Algorithm for Query-Oriented Multi-document Summarization. In: Proceeding of Information Processing (ISIP) on web mining and web-based application. IEEE, pp 9–13

Xu S, Jiang H, Lau F (2009) User-oriented document summarization through vision-based eye-tracking. In: Proceedings of the 2009 International Conference on Intelligent User Interfaces, Sanibel Island, Florida, USA. ACM, pp 7–16

Yang G, Wen D, Kinshuk Chen N, Sutinen E (2012) Personalized Text Content Summarizer for Mobile Learning: An Automatic Text Summarization System with Relevance Based Language Model. In: Proceeding of fourth international conference on technology for education, IEEE, IIIT-Hyderabad Hyderabad, India, pp 90–97