Appl Intell (2014) 40:592–600
DOI 10.1007/s10489-013-0490-0

A hybrid machine learning model for multi-document summarization

Mohamed Abdel Fattah

Published online: 20 December 2013

Abstract This work proposes an approach that uses statistical tools to improve content selection in multi-document automatic text summarization. The method uses a trainable summarizer, which takes into account several features: the similarity of words among sentences, the similarity of words among paragraphs, the text format, cue-phrases, a score related to the frequency of terms in the whole document, the title, sentence location and the occurrence of non-essential information. The effect of each of these sentence features on the summarization task is investigated. These features are then used in combination to construct text summarizer models based on a maximum entropy model, a naive-Bayes classifier, and a support vector machine. To produce the final summary, the three models are combined into a hybrid model that ranks the sentences in order of importance. The performance of this new method has been tested using the DUC 2002 data corpus. The effectiveness of this technique is measured using the ROUGE score, and the results are promising when compared with some existing techniques.

Keywords Multi-document automatic summarization · Maximum entropy · Naive-Bayes · Support vector machine

M.A. Fattah (✉)
Department of Computer Sciences, CCSE Taibah University, KSA, Almadina Almonawara, Saudi Arabia
e-mail: mohafi2003@helwan.edu.eg

M.A. Fattah
Department of Electronics Technology, FIE Helwan University, Cairo, Egypt

1 Introduction

The huge amount of information available electronically has increased demand for automatic text summarization systems. Text summarization is the process of automatically creating a compressed version of a given text that provides useful information [5, 6, 35, 40, 48]. Text summarization addresses both the problem of selecting the most important portions of text, and the problem of generating coherent summaries. There are two types of summarization: extractive and abstractive. Extractive summarization methods simplify the process by selecting a representative subset of the sentences in the original document. Abstractive summarization may compose new sentences that are not present in the original source. However, abstractive approaches require deep natural language processing techniques such as semantic representation, inference, and natural language generation, which have yet to reach a mature stage [40].

Research into automated summarization began in the 1950s [18]. Different attempts have shown that very complex techniques are required to produce human-quality text summaries because the process encompasses discourse understanding, abstraction, and language generation [34]. Simpler approaches have been explored that extract representative texts. They have used statistical techniques and/or techniques based on surface domain-independent linguistic analyses. Within this context, summarization can be defined as the selection of a subset of sentences that is representative of the document's content. This typically involves ranking the sentences in a document, so that we can select those with the highest scores and minimum overlap [4, 24]. Most recent work in summarization uses this paradigm.

The process of text summarization can be divided into three phases: analysis, transformation, and synthesis. In the analysis phase, the input text is processed and a few salient

features are selected. In the transformation phase, the results of the analysis are transformed into a summary representation. Finally, the synthesis phase produces an appropriate summary using the summary representation, which corresponds to the particular needs of a user. In the overall process, the compression rate is an important factor that influences the quality of the summary. It is defined as the ratio between the length of the summary and the length of the original text. A decreasing compression rate produces a more concise summary; however, more information is lost. An increasing compression rate produces a larger summary, but it will contain more insignificant information. In fact, when the compression rate is 5–30 % the quality of the summary is acceptable [11, 16, 20, 38].

Early work on text summarization was limited because of the lack of powerful computers and the difficulty of natural language processing (NLP), so research focused on the study of text genres such as sentence positions, and cue-phrases [7, 18]. During the 1970s, researchers began to apply artificial intelligence (AI) techniques [2, 21, 31, 41]. These AI methods exploited knowledge representations, such as frames or templates, to identify conceptual entities from a text and to extract relationships among entities by inference mechanisms. The major drawback is that limited definitions of frames or templates may lead to an incomplete analysis of conceptual entities. During the 1990s, information retrieval (IR) was used for text summarization [1, 9, 10, 15, 16, 19, 30, 36, 39]. However, most of these IR techniques focused on a symbolic-level analysis, and did not take into account semantics such as synonymy, polysemy, and term dependency [15].

Automated multi-document summarization has drawn much attention in recent years. A multi-document summary is commonly used to provide a concise topic description for a cluster of documents, and to help a user quickly browse many documents. There is an inevitable overlap in the information content of different documents. Therefore, we need effective summarization methods to merge information stored in different documents, and if possible, contrast their differences [37].

Recently, there have been many investigations into text summarization [3, 8, 32]. In [14], the authors considered the evaluation of summarization using relevance prediction, and [33] investigated the ROUGEeval package; SUMMAC, NTCIR, and DUC were considered by [28], and [13] researched voted regression models. Other techniques included single and multiple-sentence compression using "parse and trim" and statistical noisy-channel approaches [42], and conditional random fields [23]. Also investigated were multi-document summarization [12, 37] and summarization for specific domains [17, 22, 29].

In this work, all of the documents in a certain cluster have been merged into one file. After redundancy removal, the sentences of each file are modeled as vectors of features extracted from the text. The summarization task can be seen as a two-class classification problem, where a sentence is labeled as "correct" if it belongs to the extractive reference summary, or as "incorrect" otherwise. We may give the "correct" class a value '1' and the "incorrect" class a value '0'. In the testing mode, each sentence is given a value between '0' and '1' (values between 0 and 1 are continuous). Therefore, we can extract the appropriate number of sentences according to the compression rate. The trainable summarizer is expected to "learn" the patterns that lead to the summaries by identifying relevant feature values which are most correlated with the classes "correct" or "incorrect". When a new cluster file is input into the system, the "learned" patterns are used to classify each sentence by giving it a certain score value between '0' and '1'. A set of highest score sentences are chronologically specified as a file summary based on the compression rate.

This paper is organized as follows: Section 2 presents the different text feature parameters, Sect. 3 describes the proposed multi-document automatic summarization model, Sect. 4 shows the experimental results, and finally Sect. 5 presents our conclusions and future work.

2 Text features

We consider the following features when analyzing the text.

1. F1 = Word similarity among sentences.

This text feature measures the importance of a sentence based on how often its content appears in the other sentences of the document. It is simply the vocabulary overlap between this sentence and other sentences in the document. F1 is calculated as follows:

$$Score_{\mathrm{F1}}(S) = \frac{\text{Keywords in } S \cap \text{Keywords in other sentences}}{\max(\text{Keywords in } S_i \cap \text{Keywords in other sentences})} \quad (1)$$

where S is a document sentence under consideration, and S_i is sentence number i in the document. Note that the denominator of Eq. (1) is used for normalization.

2. F2 = Word similarity among paragraphs.

This text feature is similar to F1, but compares the whole paragraph rather than individual sentences. F2 is calculated as follows:

$$Score_{\mathrm{F2}}(S) = \frac{\text{Keywords in } P \cap \text{Keywords in other paragraphs}}{\max(\text{Keywords in } P_i \cap \text{Keywords in other paragraphs})} \quad (2)$$

where P is a document paragraph that contains the sentence S, and P_i is paragraph number i.

3. F3 = Text format score.

In some documents, the importance of the sentence is indicated by expressing some of its words in a different text format, e.g., italic, bold, underlined or a larger font size.

F3 is calculated as follows:

$$Score_{\mathrm{F3}}(S) = \frac{\sum_{t_{sp} \in \text{special format terms}} t_{sp}}{\sum_{t \in \text{sentence terms}} t} \tag{3}$$

where t is a phrase or term in the sentence, and t_{sp} is a sentence special format term.

4. F4 = Cue-phrases.
Sentences that contain cue-phrases such as "in summary" and "in conclusion", and superlatives such as "the best", "the most important", "according to the study", and "hardly", may be considered important.

F4 is calculated as follows:

$$Score_{\mathrm{F4}}(S) = \frac{\sum_{t_{cp} \in \text{cue-phrases}} t_{cp}}{\sum_{t \in \text{sentence terms}} t} \tag{4}$$

where t is a phrase or term in the sentence, and t_{cp} is a cue-phrase.

5. F5 = Summation of *tfidf* of the sentence terms.
The term frequency in the given document is simply the number of times a given term appears in that document. This count should be normalized to prevent a bias towards longer documents, and to give a measure of the importance of the term t_i within the particular document d_j. Thus, the term frequency is defined as follows:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}, \tag{5}$$

where $n_{i,j}$ is the number of occurrences of the considered term in document d_j, and the denominator is the number of occurrences of all terms in document d_j. The inverse document frequency is a measure of the general importance of the term and is calculated as follows:

$$idf_i \log \frac{|D|}{|\{d : t_i \in d\}|}, \tag{6}$$

where $|D|$ = the total number of documents in the corpus, and $|\{d : t_i \in d\}|$ = the number of documents where the term t_i appears (that is $n_{i,j} \neq 0$).

Then

$$tfidf_{i,j} = tf_{i,j} \times idf_i, \tag{7}$$

and

$$Score_{\mathrm{F5}}(S) \frac{\sum_t tfidf(t)}{\max(\sum_t tfidf(t) \text{ in all document's sentences})}. \tag{8}$$

6. F6 = Title feature.
A sentence is given a high score if it contains words that occur in the document title. F6 is given as follows:

$$Score_{\mathrm{F6}}(S) = \frac{\#\text{of title words in } S}{\text{title length}}. \tag{9}$$

7. F7 = Sentence location feature.
It is common for the first and last sentence of the first and last paragraphs to be important, and so it should be more likely for them to be included in the summary. F7 is calculated as follows:

$$Score_{\mathrm{F7}}(S) = \begin{cases} 1 & \text{for first or last sentence in the first} \\ & \text{or last paragraph,} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

8. F8 = Occurrence of a non-essential information feature.
Some words are indicators of non-essential information. These words are speech markers such as "because", "furthermore", and "additionally", and typically occur at the beginning of a sentence. F8 is calculated as follows:

$$Score_{\mathrm{F8}}(S) = \begin{cases} 1 & \text{if the sentence contains at least one} \\ & \text{of the non-essential information terms,} \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

3 The proposed multi-document automatic summarization model

The proposed multi-document automatic summarization model has two modes of operation:

(1) Training mode, where features are extracted from the training data and used to train the maximum entropy model, naive-Bayes classifier and support vector machine.
(2) Testing mode, where the features are calculated for the sentences in the test data. The sentences are ranked according to the sets of feature weights calculated during the training stage. We then construct a hybrid model, which is used to create the final sentence ranking. Summaries consist of the highest-ranking sentences.

3.1 Redundancy removal

A pre-processing step is necessary before the summarization process can take place. We have removed stop words and conducted some light stemming. The sentences are represented using a graph, so that we can detect and remove redundancies before applying the model-based ranking formulas. Each sentence is considered as a node in a directed graph. A link is established between two nodes if at least four continuous words are common. The link weight is the ratio of the number of common words to the average length

of the two sentences. For every parent node, those child nodes that have a link weight greater than a particular threshold are excluded from the sentence ranking process. Hence, repeated and almost identical sentences are removed.

3.2 Maximum entropy (ME)

The maximum entropy approach is appropriate for the task of sentence extraction. The maximum entropy principle encapsulates the approach of [26], which he describes as follows.

"In making inferences on the basis of partial information we must use that probability distribution which has maximum entropy subject to whatever is known. This is the only unbiased assignment we can make; to use any other would amount to an arbitrary assumption of information which by hypothesis we do not have."

The first step of the maximum entropy approach is to extract features or important information, in this case worthy sentences, to constrain the model. The second step is to construct a model using these features. As these features do not account for a complete model, we assign a uniform probability distribution, subject to the feature constraints. To find the uniform probability we must use maximum entropy. In [27], Shannon defines entropy as a measure of the information content or uncertainty of an outcome. Entropy is maximized when a distribution is uniform, i.e. this is the most uncertain situation.

The parametric form for a conditional maximum entropy model is as follows [25]:

$$P(c|s) = \frac{1}{Z(s)} \exp\left(\sum_i \lambda_i f_i(c,s)\right) \tag{12}$$

$$Z(s) = \sum_c \exp\left(\sum_i \lambda_i f_i(c,s)\right) \tag{13}$$

where c is one of two labels: one indicating that a sentence should be in the summary (correct) and another label indicating that the sentence should not be in the summary (incorrect). s is one sentence in a training set, linked to its originating document. This means that we can recover the position of any given sentence in any given document. In maximum entropy models, the training set is viewed in terms of a set of features. Each feature expresses some characteristic of the domain, as explained in Sect. 2. In Eq. (13), $f_i(c;s)$ is a feature, and λ_i is a feature's weight. We assume that all the features have the same weight.

When classifying sentences using the maximum entropy method, we use the following equation:

$$label(s) = \arg\max_{c \in C} P(c|s), \tag{14}$$

where C is a set of labels (correct and incorrect).

We can write the unnormalized score as

$$label(s) = \arg\max_{c \in C} \exp\left(\sum_i \lambda_i f_i(c,s)\right). \tag{15}$$

This maximum entropy classifier assumes a uniform prior. For a non-uniform prior, we can use

$$label(s) = \arg\max_{c \in C} F(c) \exp\left(\sum_i \lambda_i f_i(c,s)\right), \tag{16}$$

where $F(c)$ is a function equivalent to the prior when using the unnormalized classifier.

We classify each sentence using the above model.

3.3 Naive-Bayes classifier

In the naïve Bayes classifier [16], the classification function categorizes each sentence as worthy of extraction or not. Let s be a particular sentence, S be the set of sentences that make up the summary, and F1, ..., F8 the text features. Assuming that the features are independent, we get

$$P(s \in S|f_1, f_2, \ldots, f_8) = \frac{\prod_{i=1}^{8} P(f_i|s \in S).P(s \in S)}{\prod_{i=1}^{8} P(f_i)}. \tag{17}$$

Since the denominator in Eq. (17) has the same value for all sentences, it can be simplified as

$$P(s \in S|f_1, f_2, \ldots, f_8) = \prod_{i=1}^{8} P(f_i|s \in S).P(S \in S). \tag{18}$$

We assign each sentence a score using Eq. (18).

3.4 Support vector machine classifier

Support vector machine (SVM) methods have often been found to provide good classification results [3]. The SVM approach tries to find the optimal separating hyperplane between two classes.

The kernel function used to implement the SVM technique is the sigmoid function. It is

$$K(x_i, x_j) = \tanh(\gamma . x_i^T x_j + r), \tag{19}$$

where γ and r are the kernel parameters set to $\gamma = 1$.

3.5 Hybrid machine learning model

Consider a sentence, represented as a feature vector X, which is to be assigned one of n possible classes ($C_1, \ldots, C_n$). We have $n = 2$ classes, because one class indicates that a sentence should be in the summary and another class

indicates that it should not. Let R be the number of classifiers. In this case, $R = 3$ as we have the maximum entropy method, the naive-Bayes classifier, and the support vector machine classifier. The feature vector used by the ith classifier is X_i. Each class, C_k, is modeled by the probability $P(X_i|C_k)$, and its prior probability of occurrence is $P(C_k)$. The models under consideration are mutually exclusive. That implies that only one model is associated with each pattern. Using Bayesian theory, a given feature vector $X = (X_i, i = 1, \ldots, R)$ is assigned to class C_j that has the maximum posteriori probability, i.e.

$$P(C_j|X_1, \ldots, X_R) = \max_k P(C_k|X_1, \ldots, X_R). \tag{20}$$

Let us rewrite the posteriori probability, $P(C_k|X_1, \ldots, X_R)$, based on Bayes theorem as follows:

$$P(C_k|X_1, \ldots, X_R) = \frac{P(X_1, \ldots, X_R|C_k)P(C_k)}{P(X_1, \ldots, X_R)}. \tag{21}$$

The joint probability density $P(X_1, \ldots, X_R)$ can be expressed

$$P(X_1, \ldots, X_R) = \sum_{k=1}^{n} P(X_1, \ldots, X_R|C_k)P(C_k), \tag{22}$$

where $P(X_1, \ldots, X_R|C_k)$ represents the joint probability distribution extracted by the classifiers. Consider that the representations are conditionally statistically independent. Therefore, we can use

$$P(X_1, \ldots, X_R|C_k) = \prod_{i=1}^{R} P(X_i|C_k). \tag{23}$$

Substituting Eqs. (23) and (22) into Eq. (21), we get

$$P(C_k|X_1, \ldots, X_R) = \frac{P(C_k)\prod_{i=1}^{R} P(X_i|C_k)}{\sum_j^n P(C_j)\prod_{i=1}^{R} P(X_i|C_j)}. \tag{24}$$

Combining Eqs. (24) and (20), we get the decision rule. The sentence s is assigned a class, C_j, if

$$P(C_j)\prod_{i=1}^{R} P(X_i|C_j) = \max_{k=1}^{n} P(C_k)\prod_{i=1}^{R} P(X_i|C_k). \tag{25}$$

4 Experimental results

4.1 The training and testing data

We have trained our algorithm using the 147 single documents of the DUC 2001. We have extracted the eight text features and a summary from each document. We have used these feature parameters to train the models described in the previous section.

We have used multi-document extracts of 100-word summaries, generated for each of the 59 document clusters formed on the DUC 2002. First, we merged all the documents of each cluster into one file, and all of the document titles of each cluster into one title. We extracted the eight text features from each file, and then used the models to summarize the text. We ranked the sentences based on the model output. We selected a set of the highest ranked sentences for each file, with a constraint of 100 words.

We used an intrinsic evaluation to judge the quality of a summary that was based on the recall-oriented understudy for gisting evaluation (ROUGE-1). The ROUGE scores have become the standard automatic method for evaluating the content of machine generated summaries. They have been shown to be highly correlated with human evaluations. Formally, ROUGE-N (in our experiments $N = 1$) is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows:

$$\frac{\sum_{S\in\{\text{References Summaries}\}}\sum_{gram_n\in S} Count_{\text{match}}(gram_n)}{\sum_{S\in\{\text{References Summaries}\}}\sum_{gram_n\in S} Count(gram_n)}, \tag{26}$$

where n stands for the length of the n-gram, $gram_n$, and $Count_{\text{match}}(\text{gram}_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries.

4.2 Baseline approaches

4.2.1 The lead approach

We have extracted the first sentences of each document in a certain cluster to represent the cluster summary based on the 100-word constraint. Table 1 shows the average ROUGE-1 result for the 59 clusters of DUC 2002.

4.2.2 The UnifiedRank, PositionRank, TwoStageRank and BasicRank approaches

In [43], mutual influences between single-document summarization and multi-document summarization tasks are incorporated into a graph model. The ranking scores of a sentence for the two tasks were obtained in a unified ranking process. The PositionRank approach improves the basic PageRank algorithm by using the position weight of a sentence as a prior score. The TwoStageRank approach computes the score of each sentence within a single document using the PositionRank method. It then computes the final score of each sentence within the document set by considering the document-level sentence score as the prior score in

Table 1 All approach performance evaluations based on ROUGE-1

The approach	ROUGE-1	95 % Confidence Interval
Lead Baseline approach	0.2868	0.1714, 0.4022
UnifiedRank	0.3834	0.2593, 0.5075
PositionRank	0.3805	0.2566, 0.5044
TwoStageRank	0.3797	0.2559, 0.5035
BasicRank	0.3759	0.2523, 0.4995
CLASSY's guided summarization	0.3784	0.2546, 0.5022
ROUGE-1(F1)	0.3385	0.2178, 0.4592
ROUGE-1(F2)	0.3238	0.2044, 0.4432
ROUGE-1(F3)	0.2273	0.1204, 0.3342
ROUGE-1(F4)	0.2685	0.1554, 0.3816
ROUGE-1(F5)	0.3474	0.2259, 0.4689
ROUGE-1(F6)	0.3481	0.2265, 0.4697
ROUGE-1(F7)	0.2582	0.1465, 0.3699
ROUGE-1(F8)	0.2474	0.1373, 0.3575
Sum of all normalized feature parameters approach	0.3563	0.2341, 0.4785
Maximum Entropy	0.3748	0.2513, 0.4983
Naive-Bayes	0.3762	0.2526, 0.4998
Support Vector Machine	0.3813	0.2574, 0.5052
Hybrid Machine Learning Model	0.3862	0.2620, 0.5104
Hybrid Machine Learning Model using feature set (1, 2, 5 and 6)	0.3820	0.2580, 0.5060

the improved PageRank algorithm. The BasicRank approach exploits the standard PageRank algorithm to rank sentences based on all sentence relationships in a document set. Table 1 shows the ROUGE-1 results for the 59 clusters of DUC 2002 based on these four methods.

4.2.3 The CLASSY's guided summarization approach

In [46], data preparation took place before the algorithm was applied. The data preparation included sentence splitting, trimming, and categorization (do not use the sentence; use the sentence for statistics only; and consider the sentence for use in the summary). The words were stemmed, and no stop words were removed. The algorithm for sentence scoring has three parts:

- The probability that a term will be included in a human generated summary is generated for each term. The sentence score is defined as the expected number of terms in a sentence divided by the sentence length.
- A non-redundant subset of high scoring sentences is selected using non-negative matrix factorization as in [47].
- Finally, a subset of this is selected to achieve the 100-word summary using a branch and bound algorithm.

Table 1 shows the ROUGE-1 results for the CLASSY's guided summarization approach.

4.3 The effect of each feature on the summarization performance

In this section, we have investigated the effect of each feature parameter on the multi-document summarization using their score values. For instance, to investigate the effect of the first feature (word similarity among sentences) on the summarization performance, we have used $Score_{F1}(S)$ (in Eq. (1)) to rank the sentences in clusters.

Table 1 shows the summarization average ROUGE-1 result associated with each text feature.

4.4 The results using the sum of all normalized feature parameters

In this section, we have used the summation of all normalized feature parameters associated with a sentence to calculate its score value. We have used the following formula:

$$\begin{aligned} Score(S) = {} & Score_{F1}(S) + Score_{F2}(S) + Score_{F3}(S) \\ & + Score_{F4}(S) + Score_{F5}(S) + Score_{F6}(S) \\ & + \alpha Score_{F7}(S) + \alpha Score_{F8}(S) \end{aligned} \tag{27}$$

where the above equation contains the normalized feature parameters, and $\alpha = (Score_{F1}(S) + Score_{F2}(S) + Score_{F3}(S) + Score_{F4}(S) + Score_{F5}(S) + Score_{F6}(S))/6$. Table 1 shows the summarization average ROUGE-1.

4.5 The results of the maximum entropy method (ME)

The system has extracted features from the training data, which it used to train the ME model. We used the sentences of the testing data as inputs to the maximum entropy method as follows:

(1) Extract the features from the sentences in the file.
(2) Construct the feature vector.
(3) Use this feature vector as an input to the ME model.
(4) Save the output of the ME method for each sentence.
(5) Chronologically select the set of sentences based on the output of the ME model.

Table 1 shows the summarization ROUGE-1.

4.6 The results of naive-Bayes classifier

Here, we have used the steps described in Sect. 4.5, with the naive-Bayes classifier replacing the maximum entropy model. Table 1 shows the summarization average ROUGE-1.

4.7 The results of the support vector machine classifier

Here, we have used the steps described in Sect. 4.5, with the support vector machine classifier replacing the maximum entropy model. Table 1 shows the summarization average ROUGE-1.

4.8 The results of the hybrid machine learning model

We have used Eq. (25), which is a hybrid of the maximum entropy, naive-Bayes and support vector machine methods, to achieve the results in Table 1.

4.9 The results of Hybrid Machine Learning Model using the best feature set

Feature selection is a process that selects a subset of original features [44, 45]. The following algorithm uses forward feature selection (selecting the best feature at each stage).

(1) Start with a single feature and analyze the performance.
(2) Repeat Step 1 until you finish all features.
(3) Select the feature giving the best performance (now the feature set contains only one feature).
(4) Add one feature (from the rest of the available features) to the feature set then analyze the performance.
(5) Select the feature set that provides the best performance.
(6) Repeat Steps 4 and 5 until all features have been analyzed or no further performance improvement is seen.

The feature set composed of (1, 2, 5 and 6) gave reasonable results, as shown in Table 1.

4.10 Discussion

It is common for a file title to convey the main topic of its content. It is clear from Table 1 that the most important text feature for summarization is F6 (title feature), because it uses the vocabulary overlap between a sentence and the title. F5 (summation of *tfidf* of the sentence terms) also provided a good result. F1 and F2 were also effective, which is reasonable, as the sentences that contain words that appear in other sentences in the document (F1) or in other paragraphs (F2) should be important. The lowest results are associated with F7 (sentence location feature) and F8 (occurrence of non-essential information feature). The results using the support vector machine classifier are better than that of the maximum entropy and naive-Bayes approaches, as shown in Table 1. The hybrid machine learning model produced the best results, as shown in Table 1. However, it does not significantly outperform state-of-the-art approaches to multi-document summarization. Adding other text features such as positive keyword, negative keyword, Bushy path and aggregate similarity might improve the proposed method results.

5 Conclusions and future work

In this paper, we have investigated the use of maximum entropy, naive-Bayes and support vector machine models for multi-document automatic text summarization. We have also investigated a hybrid machine learning model. We have applied our new approaches to the DUC 2002 data set. Our approaches have used feature extraction criteria, which give researchers the opportunity to use many variations based on the language and text type. The text features we have used are language independent. Our achieved results are promising when compared with some existing techniques.

In the future, we will extend this approach to personalized single and multi-document summarization.

Acknowledgement This work is supported by the Deanship of Scientific Research, Taibah University, KSA.

References

1. Aone C, Okurowski ME, Gorlinsky J, Larsen B (1997) A scalable summarization system using robust NLP. In: Proceedings of the ACL'97/EACL'97 workshop on intelligent scalable text summarization, Madrid, Spain, pp 10–17
2. Azzam S, Humphreys K, Gaizauskas R (1999) Using coreference chains for text summarization. In: Proceedings of the ACL'99, College Park, MD, USA, pp 77–84
3. Begum N, Fattah M, Ren F (2009) Automatic text summarization using support vector machine. Int J Innov Comput Inf Control 5(7):1987–1996

4. Carbonell JG, Goldstein J (1998) The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st ACM SIGIR, pp 335–336
5. Diaz A, Gervás P (2007) User-model based personalized summarization. Inf Process Manag 43(6):1715–1734
6. Dorr B, Gaasterland T (2007) Exploiting aspectual features and connecting words for summarization-inspired temporal-relation extraction. Inf Process Manag 43(6):1681–1704
7. Edmundson HP (1969) New methods in automatic extracting. J ACM 16(2):264–285
8. Fattah M, Ren F (2009) GA, MR, FFNN, PNN & GMM based models for automatic text summarization. Comput Speech Lang 23(1):126–144
9. Goldstein J, Kantrowitz M, Mittal V, Carbonell J (1999) Summarizing text documents: sentence selection and evaluation metrics. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'99), Berkeley, CA, USA, pp 121–128
10. Gong Y, Liu X (2001) Generic text summarization using relevance measure and latent semantic analysis. In: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'01), New Orleans, LA, USA, pp 19–25
11. Hahn U, Mani I (2000) The challenges of automatic summarization. IEEE Comput 33(11):29–36
12. Harabagiu S, Hickl A, Lacatusu F (2007) Satisfying information needs with multi-document summaries. Inf Process Manag 43(6):1619–1642
13. Hirao T, Okumura M, Yasuda N, Isozaki H (2007) Supervised automatic evaluation for summarization with voted regression model. Inf Process Manag 43(6):1521–1535
14. Hobson S, Dorr B, Monz C, Schwartz R (2007) Task-based evaluation of text summarization using relevance prediction. Inf Process Manag 43(6):1482–1499
15. Hovy E, Lin CY (1997) Automatic text summarization in SUMMARIST. In: Proceedings of the ACL'97/EACL'97 workshop on intelligent scalable text summarization, Madrid, Spain, pp 18–24
16. Kupiec J, Pedersen J, Chen F (1995) A trainable document summarizer. In: Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'95), Seattle, WA, USA, pp 68–73
17. Ling X, Jiang J, He X, Mei Q, Zhai C, Schatz B (2007) Generating gene summaries from biomedical literature: a study of semi-structured summarization. Inf Process Manag 43(6):1777–1791
18. Luhn HP (1958) The automatic creation of literature abstracts. IBM J Res Dev 2(2):159–165
19. Mani I, Bloedorn E (1999) Summarizing similarities and differences among related documents. Inf Retr 1(1–2):35–67
20. Mani I, Maybury MT (eds) (1999) Advances in automated text summarization. MIT Press, Cambridge
21. McKeown K, Radev DR (1995) Generating summaries of multiple news articles. In: Proceedings of the 18th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'95), Seattle, WA, USA, pp 74–82
22. Moens M (2007) Summarizing court decisions. Inf Process Manag 43(6):1748–1764
23. Nomoto T (2007) Discriminative sentence compression with conditional random fields. Inf Process Manag 43(6):1571–1587
24. Nomoto T, Matsumoto Y (2001) A new approach to unsupervised text summarization. In: Proceedings of the 24th ACM SIGIR, pp 26–34
25. Nigam K, Lafferty J, Mc-Callum A (1999) Using maximum entropy for text classification. In: IJCAI-99 workshop on machine learning for information filtering
26. Jaynes ET (1957) Information theory and statistical mechanics. Phys Rev 106(4):620–630
27. Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423
28. Over P, Dang H, Harman D (2007) DUC in context. Inf Process Manag 43(6):1506–1520
29. Reeve L, Han H, Brooks A (2007) The use of domain-specific concepts in biomedical text summarization. Inf Process Manag 43(6):1765–1776
30. Salton G, Singhal A, Mitra M, Buckley C (1997) Automatic text structuring and summarization. Inf Process Manag 33(2):193–207
31. Schank R, Abelson R (1977) In: Scripts, plans, goals, and understanding. Lawrence Erlbaum Associates, Hillsdale
32. Sohrab M, Fattah M, Ren F (2008) The best feature parameter and HMM for text summarization. In: Research in computing science, and CORE-2008, 9th conference on computing, vol 34, pp 153–161
33. Sjöbergh J (2007) Older versions of the ROUGEeval summarization evaluation system were easier to fool. Inf Process Manag 43(6):1500–1505
34. Sparck Jones K (1993) Discourse modeling for automatic summarizing. Technical report 29D, Computer laboratory, University of Cambridge
35. Steinberger J, Poesio M, Kabadjov M, Ježek K (2007) Two uses of anaphora resolution in summarization. Inf Process Manag 43(6):1663–1680
36. Teufel SH, Moens M (1997) Sentence extraction as a classification task. In: Proceedings of the ACL'97/EACL'97 workshop on intelligent scalable text summarization, Madrid, Spain, pp 58–65
37. Wan X, Yang J (2006) Improved affinity graph based multi-document summarization. In: Proceedings of the human language technology conference of the North American chapter of the ACL, pp 181–184
38. Yeh J, Ke H, Yang W, Meng I (2005) Text summarization using a trainable summarizer and latent semantic analysis. Inf Process Manag 41(1):75–95
39. Yeh JY, Ke HR, Yang WP (2002) Chinese text summarization using a trainable summarizer and latent semantic analysis. In: Proceedings of the 5th international conference on Asian digital libraries (ICADL'02), Singapore. Lecture notes in computer science, vol 2555. Springer, Berlin, pp 76–87
40. Ye S, Chua T, Kan M, Qiu L (2007) Document concept lattice for text understanding and summarization. Inf Process Manag 43(6):1643–1662
41. Young SR, Hayes PJ (1985) Automatic classification and summarization of banking telexes. In: Proceedings of the 2nd conference on artificial intelligence application, pp 402–408
42. Zajic D, Dorr B, Lin J, Schwartz R (2007) Multi-candidate reduction: sentence compression as a tool for document summarization tasks. Inf Process Manag 43(6):1549–1570
43. Wan X (2010) Towards a unified approach to simultaneous single-document and multi-document summarizations. In: Proceedings of the 23rd international conference on computational linguistics, Beijing, China, pp 1137–1145
44. Brassard G, Bratley P (1996) Fundamentals of algorithms. Prentice hall, New Jersey
45. Blum AL, Langley P (1997) Selection of relevant features and examples in machine learning. Artif Intell 97:245–271
46. Conroy J, Schlesinger J, Kubina J (2011) CLASSY 2011 at TAC: guided and multi-lingual summaries and evaluation metrics. In: Proceedings of the fourth text analysis conference (TAC 2011). National Institute of Standards and Technology, Gaithersburg
47. Schlesinger J, Leary D, Conroy J (2008) Arabic/English multidocument summarization with CLASSY—the past and the future. In: Gelbukh AF (ed) CICLing, Haifa, Israel, February 2008. Lecture notes in computer science, vol 4919. Springer, Berlin, pp 568–581
48. Li J, Li L, Li T (2012) Multi-document summarization via submodularity. Appl Intell 37(3):420–430

Mohamed Abdel Fattah received the B.Sc. and M.Sc. degrees in electronics from the Faculty of Engineering, Cairo University, Cairo, Egypt, in 1994 and 2003, respectively, and the Ph.D. degree in information science and intelligent systems from the University of Tokushima, Japan, in 2007. He was awarded a Japan Society of the Promotion of Science (JSPS) postdoctoral fellowship from 2007 to 2009 in Department of Information Science and Intelligent Systems, Tokushima University. He is currently an Associate Professor with FIE, Helwan University, Cairo. His research interests include information retrieval, natural language processing, speech recognition, and document processing.