Language Resources and Evaluation
https://doi.org/10.1007/s10579-024-09739-7

ORIGINAL PAPER

Features in extractive supervised single-document summarization: case of Persian news

Hosein Rezaei[1] · Seyed Amid Moeinzadeh Mirhosseini[1] · Azar Shahgholian[2] · Mohamad Saraee[3]

Accepted: 5 April 2024

Abstract

Text summarization has been one of the most challenging areas of research in NLP. Much effort has been made to overcome this challenge by using either abstractive or extractive methods. Extractive methods are preferable due to their simplicity compared with the more elaborate abstractive methods. In extractive supervised single-document approaches, the system will not generate sentences. Instead, via supervised learning, it learns how to score sentences within the document based on some textual features and subsequently selects those with the highest rank. Therefore, the core objective is ranking, which enormously depends on the document structure and context. These dependencies have been unnoticed by many state-of-the-art solutions. In this work, document-related features such as topic and relative length are integrated into the vectors of every sentence to enhance the quality of summaries. Our experiment results show that the system takes contextual and structural patterns into account, which will increase the precision of the learned model. Consequently, our method will produce more comprehensive and concise summaries.

Keywords Supervised extractive summarization · Machine learning · Regression · Feature extraction · Natural language processing

✉ Mohamad Saraee
m.saraee@salford.ac.uk

Hosein Rezaei
hosein.rezaei@alumni.iut.ac.ir

Seyed Amid Moeinzadeh Mirhosseini
amid.moeinzadeh@gmail.com

Azar Shahgholian
A.Shahgholian@ljmu.ac.uk

[1] Isfahan University of Technology, Isfahan, Iran

[2] Liverpool Business School, Liverpool John Moores University, Liverpool, UK

[3] School of Science, Engineering and Environment, University of Salford, Manchester, UK

Published online: 08 May 2024

1 Introduction

From the early days of artificial intelligence, automatically summarizing a text has been an interesting task for many researchers. Followed by the advance of the World Wide Web and the advent of concepts such as social networks, Big Data, and cloud computing, among others, text summarization has become a crucial task in many applications (Maña-López et al., 2004; Mishra, et al., 2014; Sakai & Sparck-Jones, 2001). For example, in many search engines and document retrieval systems, it is essential to display a portion of each result entry that is representative of the whole text (Roussinov & Chen, 2001; Turpin et al., 2007). It is also becoming essential for managers and the general public to get the gist of news and articles immediately in order to save time while being inundated with information on all social media (Mckeown et al., 2005).

Researchers have approached this challenge from various perspectives and have obtained some good results (Barrera & Verma, 2012; Ferreira, et al., 2014). However, this area continues to present more research challenges and has a long path to maturity.

One method of investigating this challenge is supervised extractive summarization, which compared to unsupervised methods, is trained using labelled data. Extractive implementations use a ranking mechanism and select top-n-ranked sentences as the summary (Gupta & Lehal, 2010). Sentences of a document are represented as vectors of features. A rank will be assigned to each sentence using summarization corpora, based on their presence in golden summaries (which contain sentences of the original documents, normally selected by human). The system should then learn how to use those features to predict the rank of sentences in any given text. Various machine learning approaches such as regression and classification algorithms are used to perform the ranking task (Hirao et al., 2002; Wong et al., 2008).

As far as we know, in all current implementations, sets of sentence vectors of every document are merged to compose a larger set, which is then passed to the learning model as a matrix. In this approach, the locality of ranks is disregarded. In other words, the rank of sentences is highly relative to the context and document. A sentence might be ranked high in one document while being ranked lower in another. As a result, merging sentences of a whole dataset into a matrix removes document boundaries, and a main source of information will be lost.

We addressed this issue by taking certain features of documents into account, such as their length, topical category, and so on, and some new sentence features that also reflect document properties. Thus, although document boundaries are not explicitly reconstituted, more information will be provided to the model, and ranking could be done with respect to the local features of the document. Our experiments show that this rectification improves both the performance of the learned model and the quality of produced summaries.

We also represent a new baseline for evaluating extractive text summarizers, which can be used to measure the performance of any summarizing method more accurately.

We examined our hypothesis on a low resource language, namely Persian, which has several properties making it an appropriate case for our study. For example, a large proportion of Persian verbs are light verb constructions (Samvelian & Faghiri, 2013). In addition, due to its writing system, Persian faces relatively more homonyms, words with different meanings with the same written forms (Shamsfard, 2011). Again, context is used to determine the intended meaning.

Furthermore, in Persian, some phrases or words can be omitted if their symmetry is present in the context (Shamsfard, 2011). All of these and many other properties create challenges for summarization. As our approach is to bring the context of the document into consideration, Persian is a good candidate for our case study. However, the primary insight of this study is broadly relevant and not language-specific.

The remainder of this paper is organized as follows. Section 2 reviews related works. Section 3 presents the proposed method and evaluation measures. Section 4 discusses how the experiments are set up. The results are discussed in Sect. 5, and finally, Sect. 6 concludes the paper.

2 Related works

Both academic and industrial disciplines have widely studied text summarization. Text summarization methods may be classified into different types. Based on the input type, there are single-document (Patil et al., 2015; Torres-Moreno, 2014) vs. multi-document summarization methods (Christensen et al., 2013; Erkan & Radev, 2004; Nenkova et al., 2006), in the latter, multiple documents about a topic are summarized. Considering the language mixture, there are monolingual, bilingual, and multi-lingual methods (Gambhir & Gupta, 2017). There are also "query-focused" methods in which a summary relevant to a given query is produced (Varadarajan & Hristidis, 2006). However, from the perspective of summary generation procedure, there are two main approaches: abstractive vs. extractive (Hahn & Mani, 2000).

Abstractive approaches try to generate a new short text based on the concepts understood from the original text (Moratanch & Chitrakala, 2016). These approaches usually require a complete pass through an NLP pipeline and are faced with many complexities and challenges (Lloret & Palomar, 2012). The abstractive approach relies on linguistic methods to examine and interpret the text to find new concepts and expressions. The output is a new shorter text which consists of the essential information from the original text document (Gupta & Lehal, 2010).

On the other hand, extractive approaches select a few sentences from the document based on some measures to place them in the generated summary (Gupta & Lehal, 2010). A broad range of methods has been examined in this sector, including graph-based (Gupta & Lehal, 2010; Mihalcea & Tarau, 2004), unsupervised (Mihalcea & Tarau, 2004; Rautray & Balabantaray, 2017) and supervised (corpus-based) learning (Shafiee & Shamsfard, 2018; Silva, et al., 2015; Wong et al., 2008). In supervised methods, training data is generally needed to select important content from the documents. In these methods, the problem is usually reduced to a classification or regression problem, and machine learning techniques are applied to the dataset of documents and their gold summaries represented by some features.

Support vector machines (SVM) (Ouyang et al., 2011) and neural networks (Fattah, 2014) are more popular sentence classification algorithms.

The key step in extractive summarization is to determine the importance of sentences in the document (Fang et al., 2017). Previous studies examine the ordinal position of sentences (Edmundson, 1969), (Fattah & Ren, 2008), length of sentences (Wong et al., 2008), the ratio of nouns, the ratio of verbs, ratio of adjectives, ratio of adverbs (Dlikman, 2016), the ratio of numerical entities (Ferreira et al., 2013; Lin, 1999) and Cue words (Edmundson, 1969).

Gupta and Lehal, in their survey of text summarization techniques, list the following groups of features: content-based, title-based, location-based, length-based, proper noun and upper-case word-based, font-based, specific phrase-based, and features based on sentence similarity to other sentences in a text (Gupta & Lehal, 2010). Previous studies use different sentence features such as terms from keywords/keyphrases and user queries, frequency of words, and position of words/sentences for text summarization (Ozsoy et al., 2011).

However, in most cases, the selection and weighting of features are a crucial matter of debate. Some works have been carried out with respect to this (Neto et al., 2002), but none, to the best of our knowledge, has shown that the target attribute is highly localized within the context of the document. It is occasionally mentioned but not included in practice. For instance, Ferreira et al. studied various combinations of sentence scoring methods on three types of documents (Ferreira et al., 2013, 2014). They concluded that the weight of features varies, depending on the properties of context: "the effectiveness of sentence scoring methods for automatic extractive text summarization algorithms depends on the kind of text one wants to summarize, the length of documents, the kind of language used, and their structure". Yeh et al. (2005) utilized a Genetic Algorithm (GA) to find the weight of features for calculating sentence scores. However, their following statement implies that performance of weights is generally dependent on genre, which could be seen as a feature of context: "It cannot be guaranteed that the score function whose feature weights are obtained by GA definitely performs well for the test corpus; nevertheless, if the genre of the test corpus is close to that of the training corpus, we can make a prediction that the score function will work well." (Yeh et al., 2005). Berenjkoub et al. studied the effectiveness of various subsets of features in the summarization of distinct sections of scientific papers (Berenjkoub & Palhang, 2012). They showed that some features work well only in some specific portions of text, for example, in the abstract section, while others perform better in the methodology section. This locality effect could be considered a consequence of differences in the structure and context of each section.

All the above studies imply the significance of document context in ranking. Nevertheless, it has not been given enough attention in the NLP community and even sometimes is neglected. For instance, Dlikman (2016) suggests using a wide range of various features. Among these, seventeen part-of-speech-based sentence features have been introduced, which are all sentence-normalized, not document-normalized, i.e. they count the ratio of a syntactic unit, like verbs, divided by the number of words in a sentence. However, such features do not consider the total number of those units, e.g. verbs, in the whole document. Our work contributes to

this line of research and includes document features in the learning and ranking processes.

With regard to evaluating the results, apart from prevalent measures like ROUGE, which compare system summaries with reference summaries, there are others, such as FRESA (Saggion et al., 2010; Torres-Moreno et al., 2010) which evaluates the quality of summaries without human summaries. In our experiments, we assessed the results with and without human references using these methods.

3 Incorporating document features

As a way to investigate the need for document features in sentence ranking (as explained in the introduction and literature overview), we introduced several document-level characteristics and incorporated them into the summarization process. These features are listed under Sect. 3.1.1. Although stages of our method do not differ from state-of-the-art supervised extractive summarization, the whole process is explained to clarify and investigate the method.

Every supervised summarization has two phases. Firstly, the "Learning Phase" uses a corpus of ideal summaries to train the system to rank sentences. Secondly, in the "Summarization Phase", the system utilizes the learned model from the first phase in order to rank the sentences of a newly given text. Afterwards, the process of sentence selection is performed to form a summary of the given input. Each of these phases has several intricacies, which are briefly described in the following sections.

3.1 Learning phase

The input to this phase is a dataset of documents, each of which is associated with several human-written summaries. The output is a learned model with a good level of accuracy that is able to reliably predict the sentences' rank in almost the same way that a human might rank them. Performing normalization, sentence and word tokenization, and stop-word removal is essential. We explain the following subtasks that should be carried out later.

3.1.1 Feature extraction

Foremost, it is necessary to represent each sentence with those features that have the most distinguishing effect on the prediction of the rank. Many features have been examined in the literature. We call some "document-aware" because they implicitly represent some information about a document. However, other features that convey no information about the document have been used. We call these features "document-unaware". In the previous sections, we argued that the lack of document-related information might be misleading for the summarizer system, especially when we train it with sample sentences from different documents. Thus, we modified some document-unaware features and derived new features that cover document

properties. We also examined the effect of incorporating explicit features of a document into the vectors of its sentences. The following subsections describe the features mentioned above in more detail.

3.1.1.1 Document-unaware features *Ordinal position* It is shown that the inclusion of a sentence in a summary is relevant to its position in the document or even in a paragraph. Intuitively, sentences at the beginning or end of a text are more likely to be included in the summary as they carry more information than the body of the text. Depending on how it is defined, the position feature might be either document-unaware or not. For example, in Fattah and Ren (2008) and Suanmali et al. (2009), it is defined as 5/5 for the first sentence, 4/5 for the second, and so on down to 1/5 for the fifth and zero for the remaining sentences. Another research conducted by Wong et al. (2008) defines it as 1/sentence number. With such a definition, we may have several sentences. For instance, position = 1/5 in the training set, may not have the same sense of position. While a sentence position = 1/5 means "among the firsts" in a document with 40 sentences, it has a totally different meaning of "in the middle", for another document that contains ten sentences. Thus, a helpful feature formula should distinguish the differences between documents that may change the meaning of its information. In our experiments, we used the definition of Wong et al. (2008). Furthermore, a document-aware version of this feature will be introduced in Sect. 3.1.1.2.

Length of the sentence Intuitively, verbose or laconic sentences are less likely to be included in the summary. This is because verbose sentences undermine brevity, and laconic ones could diminish the richness of information whilst these two are prized in summarization. Like sentence position, this feature is susceptible to the misdefinition that makes it document-unaware. As an example, Wong et al. (2008) defined it as the number of words in a sentence. Such a definition does not consider the relative length of sentences to their surroundings; e.g., a sentence containing 15 words may be recognized as lengthy if other sentences in the document include fewer words.

The same sentence length might be treated as short if all other sentences in a document have more than 15 words—the root of this misinterpretation can be traced back to the writers' distinctive styles. However, we investigate this feature in our experiments to compare its effect with its document-aware counterpart, which will be listed in Sect. 3.1.1.2.

The ratio of nouns Is defined in Dlikman (2016) as the number of nouns divided by the total number of words in the sentence after stop-word removal. Three other features, ratio of verbs, ratio of adjectives, and ratio of adverbs, are defined in the same manner and have proved to have a positive impact on ranking performance. However, this feature does not capture the overall number of nouns inside a document. From our perspective, a sentence with a ratio of nouns = 0.5, for example, in a document containing many nouns, must be distinguished from another sentence in the training set with the same ratio of nouns that appeared in a document comprising fewer nouns. The same discussion justifies the need to consider the document's number of verbs, adjectives, and adverbs. The impact of these features is examined in our experiments and compared to their document-aware counterparts.

The ratio of numerical entities Assuming that sentences containing more numerical data are probably giving us more information, this feature may help us in the ranking (Ferreira et al., 2013; Lin, 1999). For calculation, it counts the occurrences of numbers and digits proportional to the length of the sentence. This feature does not take into account the numbers and digits in other sentences of the document, whereas it must be less weighted if almost all sentences of a document have numerical data. As a result, we must introduce a document-aware version of this feature.

Cue words If a sentence contains particular phrases such as "in conclusion", "overall", "to summarize", "in a nutshell", and so forth, its selection for the summary is comparatively more probable. The occurrence frequency of these phrases is calculated for this feature.

3.1.1.2 Document-aware features *Cosine position* As mentioned in Sect. 3.1.1.1, a good definition of position should reflect the length of document. A well- known formula used in the literature (Barrera & Verma, 2012; Verma & Filozov, 2010) is:

$$pos(index) = \frac{cos(\frac{2\pi \times index}{T-1}) + \propto -1}{\propto} \tag{1}$$

In which index is an integer representing the order of sentences and T is the total number of sentences in the document. This feature ranges from 0 to 1; the closer to the beginning or end a sentence is, the higher value this feature will take. Alpha is a tuning parameter. As it increases, the value of this feature will be distributed more equally over sentences. Consequently, equal values in the training set represent a uniform notion of position in a document.

Relative length The intuition behind this feature is explained in Sect. 3.1.1.1. A simple word count does not reflect the relative length of a sentence compared to other sentences that appeared in the document. In that regard, we normalize it by dividing the number of words in the sentence over the average length of sentences in the whole document. More formally:

$$Relative\ Length(s) = \frac{|s|}{\frac{\sum_1^n |s_i|}{n}} \tag{2}$$

in which n is the number of sentences in the document and s_i is the i'th sentence of it. Values greater than 1 could be interpreted as long and vice versa.

Term frequency × *inverse sentence frequency* TF-ISF counts the frequency of terms in a document and assigns higher values to sentences having more unique words. It also discounts terms that repeatedly appear in various sentences. Since it is explained thoroughly in the literature, we have not included the details and formula presented in references (Neto et al., 2000, 2002). As TF-ISF captures the frequency and inverse sentence frequency of a term with respect to its context, we can classify it as a document-aware feature.

POS features We introduce a different procedure to include the ratio of the part of speech (POS) units in features and keep them document-normalized. We divide the

number of occurrences of each POS unit by the document's total appearance instead of a sentence's total.

The formal definition of the new document-aware features is shown as follows:

$$\text{Ratio of nouns in document (s)} = \frac{number\ of\ nouns\ in\ s}{number\ of\ nouns\ in\ document} \tag{3}$$

$$\text{Ratio of verbs in document (s)} = \frac{number\ of\ verbs\ in\ s}{number\ of\ verbs\ in\ document} \tag{4}$$

$$\text{Ratio of adjectives in document (s)} = \frac{number\ of\ adjectives\ in\ s}{number\ of\ adjectives\ in\ document} \tag{5}$$

$$\text{Ratio of adverbs in document (s)} = \frac{number\ of\ adverbs\ in\ s}{number\ of\ adverbs\ in\ document} \tag{6}$$

$$\text{Ratio of numbers in document (s)} = \frac{number\ of\ numerical\ entities\ in\ s}{number\ of\ numerical\ entities\ in\ document} \tag{7}$$

3.1.1.3 Explicit document features We defined several document-specific features in order to investigate further how effective they are in sentence ranking. These features of the document will be placed in every sentence's feature vector of that document. Their formal definition is described below, and their impact is examined in the result and discussion Sect. 5:

Document sentences An essential quality of a document that affects summarization is the total number of sentences participating in sentence ranking. As this number grows, the summarization should be more selective and precise. Furthermore, some sentence features, like cue words, should be heavily weighted for longer documents. In addition, the main contextual information is distributed over sentences. Regarding this case, even nominal values of features should be considered meaningful.

Document words Another notion of document length is the number of document words. Because the number of sentences is inadequate to represent document length, we should put this feature into practice.

Topical category Different topics, such as political, economic, etc. have different writing styles, which affects sentence ranking. For instance, numerical entities appear more in economic texts or sports reports than in religious or social news. Therefore, the weight of this attribute should vary depending on a document's category.

The example in Fig. 1 represents an overview of our feature set. The ID column is for enumeration, and the Target column is explained in the next section.

		Document unaware								Explicit			Document-aware								
	ID	Ordinal position	Length	Cue words	Verb ratio in sentence	Adjective ratio in sentence	Noun ratio in sentence	Adverb ratio in sentence	Numbers ratio in sentence	Document words	Document sentences	Topical category	Cosine position	TF/ISF	Relative length	None ratio in document	Verb ratio in document	Adjective ratio in document	Adverb ratio in document	Numbers ratio in document	Target
Document 1	1	1.00	7	0	0.14	0.43	0.29	0.00	0.00	200	10	SC	1.00	7.80	0.35	0.04	0.04	0.09	0.00	0.00	0.19
	2	0.50	26	0	0.12	0.19	0.31	0.00	0.12	200	10	SC	0.88	32.92	1.30	0.16	0.11	0.15	0.00	0.50	0.30
	3	0.33	13	0	0.15	0.23	0.31	0.08	0.00	200	10	SC	0.59	17.00	0.65	0.08	0.07	0.09	0.14	0.00	0.30
	⋮																				
	8	0.13	46	0	0.20	0.13	0.24	0.02	0.02	200	10	SC	0.59	64.76	2.30	0.22	0.33	0.18	0.14	0.17	0.12
	9	0.11	18	0	0.06	0.11	0.28	0.11	0.11	200	10	SC	0.88	21.16	0.90	0.10	0.04	0.06	0.29	0.33	0.81
	10	0.10	12	0	0.08	0.17	0.17	0.00	0.00	200	10	SC	1.00	14.59	0.60	0.04	0.04	0.06	0.00	0.00	0.44
Document 2	11	1.00	17	1	0.06	0.06	0.29	0.00	0.18	590	31	SP	1.00	46.19	0.89	0.03	0.02	0.03	0.00	0.03	0.82
	12	0.50	21	0	0.05	0.10	0.24	0.00	0.19	590	31	SP	0.99	53.07	1.10	0.03	0.02	0.05	0.00	0.05	0.24
	13	0.33	5	0	0.20	0.00	0.60	0.00	0.00	590	31	SP	0.96	13.09	0.26	0.02	0.02	0.00	0.00	0.00	0.08
	14	0.25	45	2	0.11	0.07	0.40	0.00	0.07	590	31	SP	0.90	115.7	2.36	0.11	0.08	0.08	0.00	0.03	0.21
	⋮																				
	41	0.03	5	0	0.00	0.00	0.40	0.00	0.00	590	31	SP	1.00	10.98	0.26	0.01	0.00	0.00	0.00	0.00	0.25
⋮																					

Fig. 1 An excerpt of whole feature set. SC and SP under topical category stand for Science and Sport, respectively

3.1.2 Target assignment

Every feature vector needs a target value from which the system should learn how to rank sentences. The value of the target is usually determined based on golden summaries. If a sentence is included in most human-written extracts, its target is near 1. In contrast, it would be closer to 0 if the sentence could not be found in any human-made summaries. In some datasets, like Pasokh (Moghaddas et al., 2013), golden summaries are not entirely extractive, i.e. they are not composed of exact copies of sentences in the original text. Therefore, a measure of similarity between the sentences of the source text and each golden summary's sentences will be calculated, which yields real values in the range of 0 to 1. Section 4 includes more details about the target assignment.

3.1.3 Training model

Since target attribute values vary between zero and one, we opted to use regression and classification methods for the learning task. Moreover, a global matrix in which rows correspond to corpus's sentences and columns correspond to features is composed to build a training and test set. The last column shows the target attribute, which will be omitted in the test set. It might be required to perform scaling on specific columns, depending on its corresponding features and range of its values.

For large datasets, the total number of sentences that are not included in golden summaries is numerous compared to included ones. Therefore, this leads to

regression bias toward lower target values. Dataset balancing, leaving aside a portion of not included sentences and feeding the remaining to the learner model, is needed to mitigate the bias.

Lastly, the model should be fitted on the training set and be evaluated against a test set as described in Sects. 4 and 5.

3.2 Summarization phase

In this section, we briefly describe the summarization process. The evaluation process is explained in Sect. 3.3.

3.2.1 Feature extraction

Initially, sentence features need to be extracted. Normalization, sentence tokenization, word tokenization, and stop-words removal are preliminary steps. Also, the same features used in the learning phase should be calculated.

3.2.2 Sentence ranking

In comparison with the learning phase, in which a global matrix was used, a local matrix is composed whose rows correspond with the sentences of the input text. Moreover, the same scaling procedure as the learning phase should be carried out. The matrix is then fed to the regressor obtained in the previous stage to predict a rank value between zero and one for each sentence.

3.2.3 Sentence selection

The most appropriate sentences for being included in the summary will be determined by sorting sentences based on their ranks. However, it is essential to preserve original sentences order to enhance readability.

Another consideration is the cut-off length, i.e., how many of the top sentences should we select for the summary? The answer should be as simple as a constant number, a percentage of total sentences, or more advanced heuristics could determine it. We allowed cut-off length to be an input parameter, which enables us to, in the evaluation phase, produce summaries of the same length as golden summaries. Consequently, it makes the comparison more equitable.

3.3 Evaluation measures

In this section, some measures are described to evaluate the performance of both phases explained in the previous section: the learning and summarization phases. The former is evaluated using standard regression metrics such as mean square error (MSE) and coefficient of determination (R^2). The latter is carried out using ROUGE, which is a well-known metric for evaluating summarization systems.

Mean square error (MSE) is the average of squared errors in all estimated targets. An ideal regressor tends to make this measure as near as possible to zero. However, an exact zero for MSE is not desirable because it is suspected of overfitting.

The coefficient of determination is another metric for evaluating how well a regression model is fitted to data. It ranges from -∞ to 1. As it approaches 1, "goodness-of-fit" is increased, while negative values show that the mean of data is a better estimator for the target (Nagelkerke, 1991).

ROUGE is proposed in Lin (2004) as an evaluation metric for summaries. It matches n-grams in system-produced and reference summaries and returns the percentage of matches in terms of precision, recall and f-measure. A variety of ROUGE family metrics, namely ROUGE-1, ROUGE-2, and ROUGE-L, have been proposed in the literature. ROUGE-1 calculates the overlap of 1-g, ROUGE-2 the bigrams, and ROUGE-L the Longest Common Subsequence (LCS) to measure resemblance. Nevertheless, we found that ROUGE assessments are always relatively high, even for a perfunctorily produced summary. Hence, we designed a random summarizer as a baseline for comparison that selects random sentences for the summary and evaluates using ROUGE.

Evaluation without reference summaries is beneficial, especially for enormous datasets where it is impossible to get human summaries of all texts. These approaches typically compare system summaries with the documents themselves or other systems' results. Jensen Shannon Divergence (JSD) is an information-theoretic method based on the distribution of words in the original texts and system summaries. Louis and Nenkova (2009) examined several such measures and concluded that JSD is the best measure in this regard. A simple implementation of JSD is published in Ruder and Plank (2017), which we used in our evaluation.

4 Experiments

We set up two experiments to verify our hypothesis that sentence ranking is highly dependent on the document and contextual features. These experiments evaluate how effective our method, exploiting document-aware features for summarization, is against the more commonly practiced method of using document-unaware counterparts.

The first experiment involves document-unaware features (listed in Sect. 3.1.1) alongside TF-ISF. In the second experiment, document-aware features were used instead of document-unaware ones. Furthermore, we set up a random summarizer based on a random regressor that acts as a baseline for comparisons. More details are recorded in Sect. 4.4.

Moreover, we tried to find similar systems to compare our method with. However, there is hardly an available Persian summarization system comparable to ours. For example, Farahani et al. (2021) has leveraged BERT for this task, which is not comparable to our method because it's abstractive and thus, measuring overlap of their summaries with reference extractive summaries doesn't yield comparable results. We faced the same problem for many other published papers. Nevertheless,

Asgarian (2021) has revised the TextRank (Mihalcea & Tarau, 2004) algorithm and provided a web service. Thus, we used its API to compare our results with their method.

4.1 Pasokh dataset

We used the Pasokh dataset (Moghaddas et al., 2013), which contains 100 Persian news documents, each associated with five summaries. Each summary consists of several sentences of the original text selected by a human expert. Some sentences are slightly modified; therefore, they are not an exact copy of the original sentences. Pasokh's documents are categorized into six sections: political, economic, sport, science, social, and cultural, which has been reflected in the file name of documents. The length of documents ranges from 4 to 156 sentences, and it has about 2,500 sentences overall.

4.2 Extracting features and scaling

All features introduced in Sect. 3.1.1 are calculated. Pre-processing, sentence and word tokenization, stop-word removal, and part of speech tagging are performed using the Hazm library (Hazm, 2019), whose performance and effects on the process are evaluated in the 4.2.1. The list of stop words is determined from a GitHub repository.[1] After those steps, the majority of features have a range between zero and one. Other features are passed to a min–max scaler to transform into the same range. For the category feature, which is nominal, the one-hot-encoding method was applied, and six flag features were used.

4.2.1 Hazm toolkit

While English has many processing toolkits, such as NLTK and CoreNLP, Persian libraries are mostly scarce and premature. In such circumstances, the Hazm toolkit (Hazm, 2019) has proven very useful. It supports preprocessing and processing of Persian language, such as tokenization, stemming, POS tagging, dependency parsing, etc. It performs moderately well in all these tasks but stemming. Therefore, we didn't perform stemming but used the original form of words in our experiments.

4.3 Target assignment

In the target assignment, as mentioned in Sect. 3.1.2, the goal is to associate a number between 0 and 1 with higher values indicating the presence of a sentence in the majority of golden summaries. Because exact matching between sentences is not possible, to resolve the question of presence in a single golden summary such as

[1] A subset of https://github.com/kharazi/persian-stopwords/blob/master/short.

Table 1 Quality of the regression model's predictions on the test set

	MSE	R^2
Experiment 1	0.03448	0.12238
Experiment 2	0.03068	0.17576
Experiment 3, random regression	0.17112	− 3.39857

g, we calculated the cosine similarity of the desired sentence with each sentence: $s_j \in g$. Then the maximum value of these similarities is selected as an indicator of presence. This indicator is then calculated for other golden summaries, and their average is assigned to the sentence as the target:

$$Target(s) = \frac{\sum_{g_i \in G} cosine\_similarity(s, s_j)}{|G|} \tag{8}$$

G is a set of summaries written for the document containing s. This formula is additional explicit evidence that the target (and subsequently, ranking) is related to the document.

4.4 Training model

A vast collection of scikit-learn tools was used for the learning phase. K-fold cross-validation is applied with k = 4 and a split size of 0.25. Three different regression methods were applied, including Linear Regression, Decision Tree Regression, and Epsilon-Support Vector Regression. Overall results were the same, with minor differences. Thus, only the SVR result is reported. Various values for parameters were examined but the best results were achieved by epsilon = 0.01, kernel = rbf, and default values for other parameters. The fitted regressor of each run was used to rank documents' sentences in the test set to evaluate summary qualities. The produced summary should have the same number of sentences as the counterpart standard summary to have a fair comparison. Therefore, we generated system summaries, conforming to the sentence count constraint, and compared them with ROUGE. Averaging these ROUGE scores over each document and then over the dataset will indicate the overall quality of model-produced summaries.

The same process was repeated with a random regressor that needs no training and assigns a random number between zero and one to any given sample. Apart from measuring the performance of this regressor on the test set, the quality of summaries produced is evaluated and reported as a baseline. The juxtaposition of this baseline and our measured results will demonstrate how effective our feature set is and how intelligent our whole system works.

Table 2 The JSD value for each experiment

	Experiment 1	Experiment 2	Random
JSD value	0.81145	0.77567	0.79796

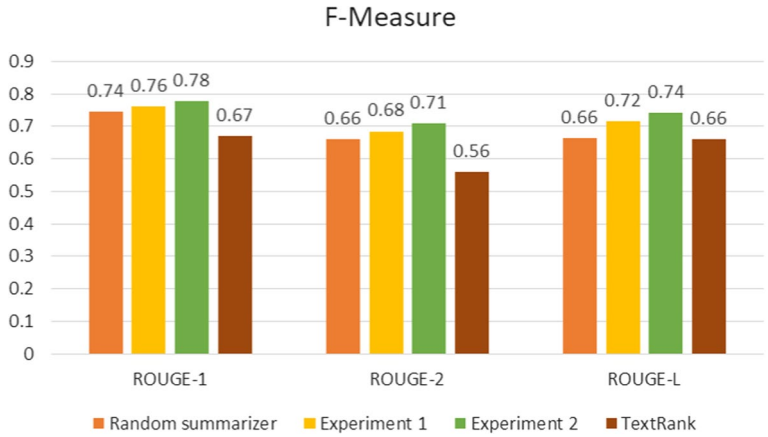

Fig. 2 ROUGE quality of produced summaries in terms of f-measure

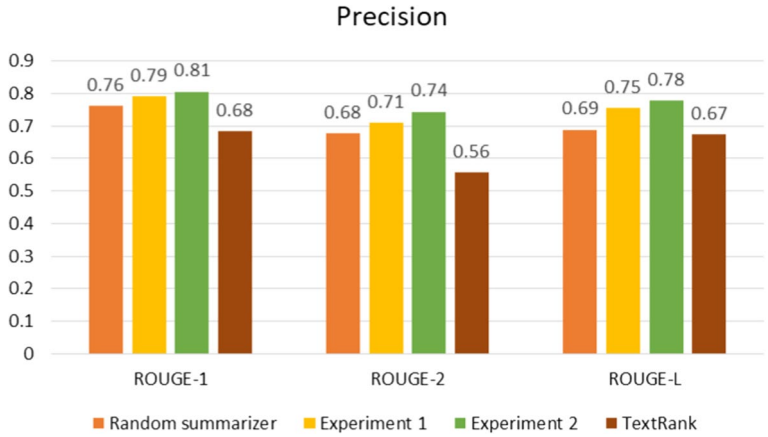

Fig. 3 ROUGE quality of produced summaries in terms of precision

5 Results and discussion

In Sect. 3.3, MSE, R^2, and ROUGE scores are noted as evaluation measures. The results of our experiments are reported below in terms of these measures. We also ran another experiment in which the random regressor was used for ranking sentences and producing summaries for better comparison. Table 1 shows and compares MSE and R^2 reported from these experiments.

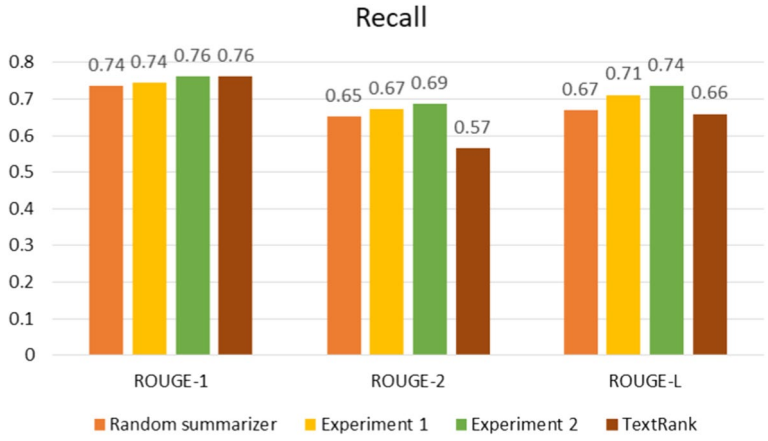

Fig. 4 ROUGE quality of produced summaries in terms of recall

The results show that in experiment 2, the mean squared error is reduced, and the R^2 score is increased. As a result, it proves that using document-aware features leads to a more accurately learned model, confirming our hypothesis about the relationship between document features and ranks.

The JSD results are displayed in Table 2. The closer the JSD value to 1 is, the better similarity has been found between the distribution of words in produced summaries and corresponding original documents. The table reveals that the first experiment showed better performance from the vantage point of evaluation without reference summaries. It can be attributed to the fact that unnecessary information might be repeated in documents and in low-quality summaries. In other words, a low-quality summary might contain repetitive information, but gain high JSD values because it mimics the same distribution of words as it is in the original texts.

Finally, the ROUGE scores are displayed separately in terms of precision, recall, and f-measure in Figs. 2, 3 and 4, respectively. F-measure scores are shown in Fig. 2, comparing ROUGE-1, ROUGE-2, and ROUGE-L. Figures 3 and 4 allow the comparison of precision and recall scores. The higher values gained in experiment 2 confirm that document-aware features perform better than unaware features.

These results are also interpretable from the viewpoint of entropy-based decision tree methods. In the learning phase, the impurity (Gini index) of features within the whole dataset will be measured, and features having higher information gain will be placed in the upper levels of the tree. But in the summarization phase, within which decisions have to be made within a single document, the impurity of those features may be low, resulting in less effective decisions and precisions. We help the model to use different features (thus different trees) for different documents by incorporating document features.

Another insight from these charts is that a random summarizer resulted in more than 50% scores in all measures. Without using document-aware features, the model achieves a slight improvement over a random summarizer.

6 Conclusion

This paper has discussed that we cannot learn to rank, in supervised extractive summarization by considering dataset sentences as independent educational examples. The rank of sentences is dependent on each other within a document. To overcome this issue, we suggested incorporating document features explicitly in a feature vector of sentences. We also suggested using features that take into account the properties of the document, document-aware features. Conducted experiments demonstrated the benefit of adding explicit document features and document-aware features, both in model precision and summary quality.

For future work, more document-aware features can be examined. For example, the position of a sentence in the paragraph seems worthy of investigation, which might be effective because paragraphs tend to have a single topic sentence and possibly a concluding sentence. They are more likely to be selected for the summary. Nevertheless, paragraph sentence position is not a reasonable choice across all languages. For instance, in Japanese, the notion of the paragraph is somehow replaced with Danraku (Kimura & Kondo, 2004), and it does not necessarily include a topic or concluding sentences.

If available, it is also possible to run the same experiments on any other language dataset. Since the features we used are based on words, sentences, and POS tags, our method is not language-specific and can be easily applied to other languages. Nonetheless, for some languages, this might not be the case. For example, in the Thai language, sentence ending markers are not explicit (Charoenpornsawat & Sornlertlamvanich, 2001). Thus, the whole idea of sentence ranking and selection faces an essential preliminary challenge of sentence tokenization, which falls beyond the scope of this paper.

Measuring the degree of entropy difference between dataset and single documents in a standard dataset can be investigated as future work. Suppose the entropy of a feature in the whole dataset is significantly different from its average entropy in each document. In that case, the feature is not applicable, and it needs interventions similar to this study.

The results of our study, though conducted before introducing Large Language Models and ChatGPT,[2] are still valid and useful. A recent study has shown that ChatGPT achieves lower performance compared to state-of-the-art extractive approaches (Zhang et al., 2023). It also has demonstrated that output of extractive methods can be used as guidance for improving the performance of ChatGPT in abstractive summarization. Thus, improvements in extractive methods are still worthy of research.

Our source code is hosted on GitHub[3] and is published for later reference, further experiments and reproducing results. A web interface[4] and a Telegram bot[5] is also implemented as a demo of our method.

[2] The main body of this study is conducted in 2017 and 2018, and the submission started on 2019, however the publication is going to happen in 2024.

[3] https://github.com/Hrezaei/SummBot.

[4] http://parsisnlp.ir/summ/form.

[5] https://t.me/Summ_bot.

Declarations

Conflict of interests The authors declare that they have no conflict of interests.

References

Asgarian, E. (2021). "Text-Mining.ir," [Online]. Available: https://demo.text-mining.ir/Home/Summarization. Accessed: 03-May-2021.

Barrera, A., & Verma, R. (2012). Combining syntax and semantics for automatic extractive single-document summarization. In CICLing'12 Proceedings of the 13th International Conference on Computational Linguistics and Intelligent Text Processing—Volume Part II, pp. 366–377.

Berenjkoub, M., & Palhang, M. (2012). Persian text summarization using a supervised machine learning approach. In Proceedings of the Robocup IranOpen 2012 Symposium and 2nd Iran's Joint Conference of Robotics and AI, Tehran, Iran.

Christensen, J., Soderland, S., & Etzioni, O. (2013). Towards coherent multi-document summarization. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1163–1173.

Dlikman, A. (2016). Using machine learning methods and linguistic features in single-document extractive summarization. DMNLP@PKDD/ECML, pp. 1–8.

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

Erkan, G., & Radev, D. (2004). LexPageRank: Prestige in multi-document text summarization. EMNLP, pp. 365–371.

Fang, C., Mu, D., Deng, Z., & Wu, Z. (2017). Word-sentence co-ranking for automatic extractive text summarization. *Expert Systems with Applications, 72*, 189–195.

Farahani, M., Gharachorloo, M., & Manthouri, M. (2021). Leveraging ParsBERT and pretrained mT5 for Persian abstractive text summarization. In 2021 26th International Computer Conference, Computer Society of Iran (CSICC), pp. 1–6. IEEE.

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

Fattah, M., & Ren, F. (2008). Automatic text summarization. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2*(1), 90–93.

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

Ferreira, R., Freitas, F., Cabral, L., Lins, R., Lima, R., & Franca, G., et al. (2014). A context based text summarization system. In 2014 11th IAPR International Workshop on Document Analysis Systems, pp. 66–70.

Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review, 47*(1), 1–66.

Gupta, V., & Lehal, G. (2010). A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence, 2*(3), 258–268.

Hahn, U., & Mani, I. (2000). The challenges of automatic summarization. *IEEE Computer, 33*(11), 29–36.

Hazm. (2019). Retrieved from GitHub—sobhe/hazm: Python library for digesting Persian text. https://github.com/sobhe/hazm

Hirao, T., Isozaki, H., Maeda, E., & Matsumoto, Y. (2002). Extracting important sentences with support vector machines. In COLING '02 Proceedings of the 19th international conference on Computational linguistics—Volume 1, pp. 1–7.

Kimura, K., & Kondo, M. (2004). Effective writing instruction: From Japanese danraku to English paragraphs. In Proceedings of the 3rd Annual JALT Pan-SIG Conference, pp. 22–23.

Lin, C.-Y. (1999). Training a selection function for extraction. In Proceedings of the Eighth International Conference on Information and Knowledge Management, pp. 55–62.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In Text Summarization Branches Out: Proceedings of the ACL-04 Workshop, pp. 74–81.

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

Louis, A., & Nenkova, A. (2009). Automatically evaluating content selection in summarization without human models. In Conference on Empirical Methods in Natural Language Processing.

Maña-López, M., Buenaga, M., & Gómez-Hidalgo, J. (2004). Multidocument summarization: An added value to clustering in interactive retrieval. *ACM Transactions on Information Systems, 22*(2), 215–241.

Mckeown, K., Passonneau, R., Elson, D., Nenkova, A., & Hirschberg, J. (2005). Do summaries help. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 210–217.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In Conference on Empirical Methods in Natural Language Processing (EMNLP), 2004, Barcelona, Spain, pp. 404–411.

Mishra, R., Bian, J., Fiszman, M., Weir, C., Jonnalagadda, S., Mostafa, J., & Fiol, G. (2014). Text summarization in the biomedical domain. *Journal of Biomedical Informatics, 52*, 457–467.

Moghaddas, B., Kahani, M., Toosi, S., Pourmasoumi, A., & Estiri, A. (2013). Pasokh: A standard corpus for the evaluation of Persian text summarizers. ICCKE 2013, pp. 471–475.

Moratanch, N., & Chitrakala, S. (2016). A survey on abstractive text summarization. In 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), pp. 1–7.

Nagelkerke, N. (1991). A note on a general definition of the coefficient of determination. *Biometrika, 78*(3), 691–692.

Nenkova, A., Vanderwende, L., & Mckeown, K. (2006). A compositional context sensitive multi-document summarizer: Exploring the factors that influence summarization. In Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 573–580.

Neto, J., Freitas, A., & Kaestner, C. (2002). Automatic text summarization using a machine learning approach. In Brazilian Symposium on Artificial Intelligence, pp. 205–215.

Neto, J. L., Santos, A. D., Kaestner, C. A., & Freitas, A. A. (2000). Document clustering and text summarization. In Proceedings of the 4th International Conference Practical Applications of Knowledge Discovery and Data Mining (pp. 41-55). The practical application company.

Ouyang, Y., Li, W., Li, S., & Lu, Q. (2011). Applying regression models to query-focused multi-document summarization. *Information Processing and Management, 47*(2), 227–237.

Ozsoy, M., Alpaslan, F., & Cicekli, I. (2011). Text summarization using latent semantic analysis. *Journal of Information Science, 37*(4), 405–417.

Patil, A., Pharande, K., Nale, D., & Agrawal, R. (2015). Automatic text summarization. *International Journal of Computer Applications, 109*(17), 18–19.

Rautray, R., & Balabantaray, R. (2017). An evolutionary framework for multi document summarization using Cuckoo search approach: MDSCSA. *Applied Computing and Informatics, 14*(2), 134–144.

Roussinov, D., & Chen, H. (2001). Information navigation on the web by clustering and summarizing query results. *Information Processing and Management, 37*(6), 789–816.

Ruder, S., & Plank, B. (2017). Learning to select data for transfer learning with Bayesian optimization. In Conference on Empirical Methods in Natural Language Processing.

Saggion, H., Torres-Moreno, J., Cunha, I.D., SanJuan, E., & Velázquez-Morales, P. (2010). Multilingual summarization evaluation without human models. In International Conference on Computational Linguistics.

Sakai, T., & Sparck-Jones, K. (2001). Generic summaries for indexing in information retrieval. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 190–198.

Samvelian, P., & Faghiri, P. (2013). Introducing PersPred, a syntactic and semantic database for Persian complex predicates. In The 9th Workshop on Multiword Expressions, pp. 11–20.
Shafiee, F., & Shamsfard, M. (2018). Similarity versus relatedness: A novel approach in extractive Persian document summarisation. *Journal of Information Science, 44*(3), 314–330.
Shamsfard, M. (2011). Challenges and open problems in Persian text processing. *Proceedings of LTC, 11*, 65–69.
Silva, G., Ferreira, R., Lins, R., Cabral, L., Oliveira, H., Simske, S., & Riss, M. (2015). Automatic text document summarization based on machine learning. In Proceedings of the 2015 ACM Symposium on Document Engineering, pp. 191–194.
Suanmali, L., Salim, N., & Binwahlan, M. (2009). Fuzzy logic based method for improving text summarization. arXiv preprint http://arxiv.org/abs/0906.4690.
Torres-Moreno, J.-M. (2014). Automatic text summarization: Torres-Moreno/automatic text summarization.
Torres-Moreno, J., Saggion, H., Cunha, I. D., SanJuan, E., & Velázquez-Morales, P. (2010). Summary evaluation with and without references. *Polibits, 42*, 13–19.
Turpin, A., Tsegay, Y., Hawking, D., & Williams, H. (2007). Fast generation of result snippets in web search. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 127–134.
Varadarajan, R., & Hristidis, V. (2006). A system for query-specific document summarization. In Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 622–631.
Verma, R., & Filozov, F. (2010). Document Map and WN-Sum: A new framework for automatic text summarization and a first implementation. Technical Report UH-CS-10-03, University of Houston Computer Science Dept.
Wong, K.-F., Wu, M., & Li, W. (2008). Extractive summarization using supervised and semi-supervised learning. In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pp. 985–992.
Yeh, J.-Y., Ke, H.-R., Yang, W.-P., & Meng, I.-H. (2005). Text summarization using a trainable summarizer and latent semantic analysis. *Information Processing and Management, 41*(1), 75–95.
Zhang, H., Liu, X., & Zhang, J. (2023). Extractive summarization via chatgpt for faithful summary generation. arXiv preprint http://arxiv.org/abs/2304.04193