Wajeez: An Extractive Automatic Arabic Text Summarisation System

Abrar Al Oudah[1], Kholoud Al Bassam[1], Heba Kurdi[1], and Shiroq Al-Megren[2(✉)]

[1] Computer Science Department, King Saud University, Riyadh, Saudi Arabia
{436203175,437202894}@student.ksu.edu.sa, hkurdi@ksu.edu.sa
[2] Information Technology Department, King Saud University, Riyadh, Saudi Arabia
salmegren@ksu.edu.sa

Abstract. The volume of Arabic information is rapidly increasingly nowadays, and thus, access to the corrects is arguably one of the most difficult research problems facing readers and researchers. Text Summarisation Systems are utilised to produce a short text describing significant portions of the original text. That is by selecting the most important sentences, following several steps: preprocessing, stemming, scoring, and summary extraction. Nevertheless, summarisation systems remain still in their infancy for the Arabic language. Therefore, this paper proposes an automatic Arabic text summarisation systems, entitled Wajeez, that introduces a new inclusive scoring formula that generates a final summary from several top-ranking sentences. Wajeez was applied on two different datasets: the Essex Arabic Summaries Corpus (EASC) and a manual summary to assess its performance using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) set of metrics. In comparison to two other competitions systems, Wajeez performed comparatively well when a title is provisioned to support summarisation.

Keywords: Natural language processing · Text summarisation · Extractive approach · Arabic language · Sentence scoring methods · Ranking

1 Introduction

As research on text summarisation is becoming a hot topic in Natural Language Processing. There is a big demand for automatic text summarisation systems which automatically retrieves the data from documents that minimising our precious time. The benefits of automatic Arabic text summarisation appear on some applications and fields such as the first page of the newspaper is a brief summary of next pages, sending news by SMS that helps in keeping the time of users. Eduard Hovy defines a summary as: "a text that is produced from one or more texts, which contains a significant portion of the information in the original text(s), and that is no longer than half of the original text(s)" [13].

G. Meiselwitz (Ed.): HCII 2019, LNCS 11578, pp. 3–14, 2019.
https://doi.org/10.1007/978-3-030-21902-4_1

Text summarisation can be classified according to different criteria; one of these criteria is summarisation method. The summarisation methods can be classified into abstractive and extractive summarisations [11]. Extractive Text Summarisation (ETS) uses classical approaches to generate summaries by cropping important segments of the original text and combining them to build a consistent summary. In Abstractive Text Summarisation (ATS), the document will be generated from scratch without being restrained to phrases from the original text. This is like the human-written sentences to generate summaries.

The Arabic language is the native language for more than 300 million people worldwide and one of the six official languages used at the United Nations. The Arabic languages includes 28 letters and is written from right to left. The letters change forms according to their position in the word. Furthermore, Arabic short vowels do not appear as letters it appears as diacritical marks, which are marks written on the top. In addition, the Arabic language has no capitalisation. In written Arabic, if the vowels omitted then the result will have a higher level of ambiguity. This ambiguity will be a problem in information retrieval, in the fact that an Arabic word can have several meanings. In addition to the ambiguity, there is another problem of the plural form of irregular nouns, also called broken plural. In this case, a noun in plural takes another morphological form different from its initial form in singular [16].

There is a need of Arabic text summarisation as the number of texts written in Arabic rapidly increases. Users cannot manually handle a large amount of text, and thus there is a necessity to have an automated system to generate a summary to give the reader the main idea of the text. Therefore, the need to automate Arabic text summarisation was a desirable goal. Text summarisation (TS) aims to generate a new document (summary) which consisting of a few sentences capture the most important content of an input document or a set of documents.

This paper proposes a new system, Wajeez, that automatically generates summarisation of Arabic text to balance both information scores and readability. That is by extracting sentences of the highest scores to help understand the general meaning of the original text. The main objectives of this work is examine the effect of combining many features of text extraction that generate an automatic Arabic summary text depending on the extraction of the most salient sentences of a text that help to understand the general meaning of the whole text. The performances of Wajeez is comparatively assessed against two established systems using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) set of metrics. The findings support the usefulness of Wajeez at producing valid summarisation when an adequate title is provided with the text to be summarised.

The remainder of this paper is organised as follows. First, the related work section reviews work for extractive text summarisation system for both Arabic and English languages. Third, the following sections describes Wajeez, the proposed extractive Arabic text summarisation systems. Fourth, the experimental

set-up is descried and the results are presented and discussed. The final section summarises and concludes the paper and briefly draws future directions.

2 Related Work

The past few years have seen many works proposing text summarisation techniques for various languages. In this section, a review of the literature on extractive text summarisation system for English and Arabic language will be presented. First, we will present many related works for the English language. Then some Arabic text summarisation system will be shown in the second part.

Since 1950's, a lot of researches have been conducted on text summarisation using a variety of and languages. For an English language system, a computer program that used machine learning and natural language processing developed approaches to automatically generate summaries of full-text scientific publications [8]. This program uses an extractive approach to generate the summery. The summaries of the sentence and fragment levels were evaluated in finding common clinical systematic review (SR) data elements.

A novel word-sentence co-ranking model named CoRank was proposed [12]. This CoRank model combines sentence scoring techniques by combine the word-sentences relationship with the graph-based unsupervised ranking model. The actions of CoRank are also supplemented with a redundancy elimination techniques. Two real-life datasets were used to assess the performance of CoRank with nearly 600 documents. The findings confirm the effectiveness of the proposed approach in producing short summaries.

A model for automatic text summarisation was introduced and was based on fuzzy logic system evolutionary algorithms and cellular learning automata [1]. First, the proposed approach extracts the most important features. A linear combination of these features shows the importance of each sentence. A combined method based on artificial bee colony algorithm and cellular learning automata are then used to calculate similarity measures. Furthermore, a new method is proposed to set the best weights of the text features using particle swarm optimisation and genetic algorithm. This method assigns weights to all text by discovering more important and less important text features. In the end, a fuzzy logic system is used to perform the final scoring.

A topic modeling approach to extractive automatic summarisation was presented to achieve a good balance between compression ratio, summarisation quality, and machine readability [18]. The approach goes through many steps, the first step is extracting the candidate sentences associated with topic words from a preprocessed novel document. Then, select the most important sentences from the candidate sentences to generate an initial novel summary. The last step is the summary smoothing to improve the summary readability by overcomes the semantic confusion caused by ambiguous or synonymous words.

Text summarisation field has not been studied enough for Arabic language and currently, only a few related works are available. The first system designed for Arabic text summarisation, uses a combination of many statistical features

(terms frequency, the position of the sentence, etc.) [9]. Extractive Arabic Text Summarization based on graph theory and semantic similarity between sentences to calculate importance of each sentence in the document so to extract the most important sentence was also developed [6]. For a final extractive summary, they used the ranking algorithm in combination with Maximal Marginal Relevance method instead of selecting top-ranked sentences. For experimentation, they had been building their own corpus of Arabic articles, a total number of documents is 25 were the summaries produced manually by an expert in the Arabic language. Their approach evaluated using Precision, Recall, and F-measure and obtained by the following values 79.0%, 71.8% and 75.22% respectively.

An Arabic text summarisation system that combines statistical and semantic approaches to achieve the summarisation task implemented was more recently proposed [5]. The system goes through three steps; preprocessing, computes P the similarity between each pair of sentences, then converts the text into a graph model using PageRank algorithm. Then, the score of each sentence is improved by adding other statistical features such as TF.IDF and sentence position. Including the top-ranked sentences from the input, document forms the summary, and an adapted version of maximal marginal relevance (MMR) algorithm is applied to remove information that is redundant and enhance the quality of the result summary.

Text summarisation relies heavily on human-engineered features. Therefore, a generic extractive Arabic Single-Document summarisation approach using the hybrid graph and statistical approaches integrated with the Genetic Algorithm that depends on the semantic relationship between sentences was proposed [14]. The evaluation of this approach using EASC corpus, and ROUGE evaluation method to determine the accuracy. With compression ratio equal to 40% of the original text size, the F-measure equal to 0.5476 for ROUGE-1 and 0.4465 for ROUGE-2. In another paper, an approach for using Practical Swarm Optimisation (PSO) algorithm for summary extraction for single Arabic document was produced [2]. In the research, EASC corpus used as a dataset and evaluated the result using ROUGE tool. PSO approach combined informative scoring with semantic scoring to find the optimal summary.

In conclusion, it is notable that each of the above-mentioned approaches has its own advantages towards single-document summarisation. However, there are some issues and limitation. From this section, we can see that there is no approved benchmark for the Arabic language used in the evaluation process neither gold standard corpora and the different measures used to assess text summarisation. This gap is address in this works and its exploration of appropriate techniques for the Arabic text summarisation.

3 Extractive Arabic Text Summarisation System

In this section, the processes and structure of our proposed Extractive Automatic Arabic Text Summarization System, Wajeez, are presented. The system is easy to use and has the feature of enabling the user to determine the size of the

summary through a number of sentences, a number of words and a percentage of the original. It also allows the user to enter a query to enhance the summary. i.e. the user can enter a specific word or a list of words. First, we divide the text to set of sentences. We then assign a score to each sentence based on seven scoring functions calculated for each sentence. At the end, we pick sentences that have a maximum score while satisfying the constraint set by the user in terms of length of the summary and the query if it had entered Otherwise the system will generate the default summary size which is 30% as some other system [5,6]. Wajeez processes on five main subsystems: data acquisition, preprocessing, stemming, sentence scoring, and finally we generate the summary within the user determined size. The following subsections summarises each subsystems, with a particular concentration on the scoring function.

3.1 Data Acquisition

The data acquisition subsystem describes the inputs of the system and it is required structure. Text, query, and size of the text are the main inputs of the system. The user then selects either support the system with a query to satisfy a requested summary, which would be a sentences, word, or a collection of words. The user would then specify the size of the summary as a generated ratios, a specified number of sentences or number of words.

3.2 Preprocessing

The preprocessing subsystem is a structured representation of the original text. Preprocessing has six main steps: sentence segmentation, tokenisation, removing sentence noise word, and removing stopwords, and extracting strong words. The task of preprocessing is to segment a text into a set of sentences then filter the sentences from undesirable additions such that: numbers, symbols, punctuation, diacritics, define article and stop words.

Sentence Segmentation. Texts segmentation is the process of dividing input text into meaningful units called a sentence. This task involves identifying sentence boundaries between words in different sentences [15]. Most written languages, including Arabic language, have main punctuation marks which occur at sentence boundaries such as the period, question mark, and exclamation mark. One sentence segmentation issue occurs with a period, where it can be used for terms such as Dr. in the English language. To deal with this issue we proposed a list of Arabic abbreviations that end with or contain period, which contributed to an increased level of performance.

Tokenisation. Tokenization is a text term number which is a count measure used in calculating a number of words in text input to be text's size. It is used in generating a summary if a user specified a desired size of summary by words number. In addition, it is important in the coming preprocessing steps, which deals with each word individually.

Removing Sentence Noise Word. The noise of sentence means the additions to a sentence or to individual words that not needed in next processes. It includes filtering each sentence from numbers, punctuations, and diacritics and defines articles.

Removing Stopwords. Stop words are non-meaning words appear mostly in the text to conjunction other words. There are categories of stop words: prepositions, pronouns, relative pronouns, demonstrative pronouns, etc. Stop words filtered out before or after processing of natural language data to make the process easier. In Wajeez, an open source list of Arabic stopwords is used [7].

Extracting Strong Words. Strong words in Arabic are word with a strong meaning that needs to be considered when calculating the scoring function to increase the importance of a sentence that appears with one or more strong words. It is a statistical technique to identify and increase the strong word counter value if one of the strong words appears in a sentence.

3.3 Stemming

The stemming subsystem is converting each word to its stem, which the past verb forms. All words are derived from the stem, and there are several kinds of derives: verb derivation, noun derivation, and infinitive derivation. Summarisation systems need to stem process because it works with the meaning of the words, not the forms, the system then will calculate the frequency of stems and compute the similarity of two sentences, the similarity with title or query if it exists. In Wajeez, three stemmers were investigated: Khoja, ISRI, and Tashaphyne light stemmer.

3.4 Sentence Scoring

In the process of identifying important sentences, determining the factors that affect the relevance of sentences. Our proposed system uses the following seven factors to calculate scoring function for each sentence: Position (Ps) also known as the Location of Sentence, Length of Sentence (L), Frequencies (F) for each word on sentence, Similarity (SI) to determine whether this sentence related to other sentences, existence of Strong Word that commonly used on Arabic (SW), existence of Title Matching Stems (TM) and user entered Query Matching Stems (QM).

To analysis scoring function first, we had tested the factors separately on some text to calculate how much each factor affects summarisation by using standard evaluation measures for each factor. Compute the factors for each sentence by using following equations.

Let Ps be a position of the sentence that calculated using a counter that retains the appearance position of each sentence in the original text. Then:

$$Ps(s(i)) = \frac{Ns - Ps}{Ns} \tag{1}$$

Where Ns is the number of sentences in original text and P is the position of sentence $s(i)$.

Let L be the length of sentence which computed by calculating the number of stems in a sentence $s(i)$ divided by the number of words in the sentence that has the highest length.

$$L(s(i)) = \frac{Number\ of\ stems}{Number\ of\ words\ of\ the\ highest\ length\ of\ sentence} \tag{2}$$

The frequency of a word or word frequency Fw is count repetition of each word individually in a whole text. Then divide by total text words.

$$Fw(w(i)) = \frac{count(w(i))}{Nw} \tag{3}$$

Where $count(w(i))$ is the number of occurrence of word $w(i)$ and Nw is the total number of words in the document. Then, frequency F for each sentence $s(i)$ is calculated below where k is the number of words in sentence $s(i)$.

$$F(s(i)) = \frac{\sum_{i=1}^{k} Fw(w(i))}{k} \tag{4}$$

Once you are reading the title you will get the idea of that text. Sentence s(i) will be classified as important if there exists intersection with title stems. The similarity with the title will be calculated by:

$$TM(s(i)) = \frac{stem(s(i)) \cap stem(t)}{stem(s(i)) \cup stem(t)} \tag{5}$$

Where $stem(s(i))$ is the stem of words in a sentence $s(i)$ and $stem(t)$ is the stem of words in the title t.

The sentence $s(i)$ is more important than other sentences if its stems commonly appear in other text sentences. The similarity with text sentences will be calculated below where $stem(ts)$ represent words stems of all text sentences and $stem(s(i))$ be as described previously.

$$SI(s(i)) = \frac{stem(s(i)) \cap stem(ts)}{stem(s(i)) \cup stem(ts)} \tag{6}$$

Hence this system is built to serve user as much as possible, it asks the user to enter a query that includes desirable words or complete sentence which the user prefer to appear on the generated sentences. Query matching will be calculated for sentences $s(i)$ as:

$$QM(s(i)) = \frac{stem(s(i)) \cap stem(q)}{stem(s(i)) \cup stem(q)} \tag{7}$$

Where $stem(s(i))$ is the stem of words in a sentence $s(i)$ and $stem(q)$ is the stem of words in the user entered a query.

The Arabic language includes some words that make the sentence a strong meaning which are preferable in summary. The sentence $s(i)$ which has strong words will be assigned a score by:

$$SW(s(i)) = \frac{Number\ of\ strong\ word\ in\ s(i)}{length\ of\ sentence\ s(i)} \tag{8}$$

After calculating a scoring function for each sentence using the equations above, the sentences should be sorted according to its scoring function value and its weight in a decreasing order. The proposed scoring function is a function based on the linear combination of all factors, then multiplying the score for each sentence by a scale value to normalise fluctuated factor values of each factor to improve the summary extracted result.

$$Score(s(i)) = Ps(s(i)) + L(s(i)) + F(s(i)) + SI(s(i)) + TM(s(i)) + QM(s(i)) \tag{9}$$

$$S(i) = \sum_{k=1}^{N} S(k) \cdot Score(s(i)) \tag{10}$$

Where $S(k)$ is the scale to normalise fluctuated feature values of each factor and N represents the number of factors and i is represented the number of sentences which entered by the user. To calculate this scale, take the average values of factor F respect to all sentences and repeat this with all factors. Pick the largest average of these factors then divides it by factor value as:

$$S = \frac{Largest\ average}{Average\ factor\ value} \tag{11}$$

3.5 Summary Generation

This is the last step in our system, which produce the output (summary) to the user according to the entered size either by ratio, number of sentence or number of words. If the user chooses to summarise the text based on "sentences number", then the extraction process done by ranking the sentences based on their scores, after that extract the highest ith sentences. If the size unit is ratio or number of words Wajeez system using LengthController algorithm which is based on the 0/1-Knapsack problem to generate the final summary which ensures it has a highest possible score and not exceed the size [10]. Finally, order these extracted sentences depending on the priority of their position as they appear in the original text.

4 System Evaluation

Wajeez was developed using Python 2.7.x programming language. In our system, we used two datasets to evaluate our system: EASC and manual summaries. EASC is used to evaluate the proposed approach. The corpus includes

153 documents and has five summaries for each document; with a total of 765 Arabic human-made summaries generated using Mechanical Turk (Mturk) [10]. Ten subjects are embedded in EASC corpus: art, music, environment, politics, sports, health, finance, science and technology, tourism, religion, and education. In Wajeez System, we select 3 articles from each subject. Manual summaries we provide 12 texts from different sources: sites, newspapers, and e-book that are in different scopes type: science, news, research, general Articles, hadith, and health.

Evaluating the summarisation approaches face a challenge with Arabic text summarisation systems, it can be done manually, automatically, or semi-automatically [4]. However, it appears that one of the main problems in Arabic text summarisation is the absence of Arabic gold standard summaries. Moreover, the difficulties of evaluation for Arabic summarisation is due to the lack of Arabic benchmark corpora, lexicons and machine-readable dictionaries make automatic [3].

In the evaluation, we will measure the actual quality of the machine summaries generated by our system. The summary will be produced based on one of three units either percentage, a number of sentences or number of words that user specify it. We will combine two based of the summary techniques: Generic and Query to improve the quality of the machine summary.

We ran our algorithm to generate summaries for sample texts in different sizes: 20%, 25%, 30%, 35%, and 40% which are the ratios of the summary length to the original document length. To evaluate the system generated summaries, three measures were used: precision, recall, and F-measure. F-measure (F) balances recall and precision using a parameter β, where $\beta = 1$. This means precision and recall are given equal weight. More formally, we assume that S_{manual} is the set of sentences in the manual summary and S_{auto} is the sentences in the auto-generated summary, therefore:

$$P = \frac{|S_{manual} \cap S_{auto}|}{S_{auto}} \tag{12}$$

$$R = \frac{|S_{manual} \cap S_{auto}|}{S_{manual}} \tag{13}$$

$$F = \frac{2PR}{P + R} \tag{14}$$

Table 1 displays the performance results of Wajeez using the EASC dataset at varying summary sizes. We then compare our system with two competitive approaches, Arabic Summarisation based on Graph Theory (ASGT) and Arabic Summarisation based on the statistical and semantic analysis. ROUGE toolkit [7,17] had been used to evaluate our automatically generated summaries that are a widely used evaluation metric. We choose to take the summary size equal to 30% in the comparison. The results of comparing our system with the other two systems are shown in Table 2.

Table 1. Wajeez's evaluation results on the EACS dataset at varying summary sizes

Summary size	F-measure
20%	0.27
25%	0.29
30%	0.32
35%	0.33
40%	0.34

Table 2. Wajeez's evaluation results on the EACS dataset in comparison with ASGT and SSAS.

Summary size	F-measure
ASGT	46.75
SSAS	58.2
Wajeez	31.5

Table 2 shows that Wajeez has the lowest performance over the other two systems. We conclude that this may occur because our system has the best performance when the text contains a title. SSAS system has the highest F-measure with 58.2%.

On the other hands, we had tested the quality of the machine summary using the manual dataset. We provide 12 texts from different sources (Sites, Newspapers, and E-book that are in different scopes type: Science, News, Research, General Articles, Hadith and Health) to an expert to get human summaries and experience of these texts. Then we auto-generate the summary for these texts using Wajeez system in different sizes. The summaries were evaluated to get Precision (P), Recall (R) and F-measure (F) measures for untitled text and title text as well. Table 3 shows the average results of the manual summary results. After evaluating many tests, the system performance with F measures is equal to 0.59 with title feature calculation and 0.58 without title calculation, which considered as a high performance compared to some other Arabic systems. In addition, Wajeez system gives greater value for its performance at 40%, and this performance will increase up when percentage increases.

Table 3. Wajeez's evaluation results on the manual text summary

Summary size	P	R	F-measure
15%	0.422	0.422	0.422
25%	0.551	0.678	0.607
30%	0.583	0.582	0.578
40%	0.630	0.600	0.639
15% (with title)	0.472	0.500	0.482
25% (with title)	0.644	0.737	0.687
30% (with title)	0.575	0.617	0.593
40% (with title)	0.676	0.753	0.710

5 Conclusion and Future Work

This paper proposed a new Extractive Arabic text Summarisation, entitled Wajeez. The main contribution of this work is the investigation of a preprocessing methods, preprocessing for a light stemmer Tashaphyne [16] to enhance its performance and scoring function. Testing was done to measure the performance of the system for the EASC corpus dataset as well as the manual dataset. Rouge toolkit was used, in addition to EASC corpus; the system has a low performance comparing with ASTG and SSAS with F-measure equal to 31.5% for our system and 46.75% and 58.2% respectively. After perform testing on the manual dataset, the performance of the system with F-measures is equal to 59% with title feature calculation and 58% without title calculation. In addition, Wajeez system gives greater value for its performance at 40%, and this performance will increase up when parentage increases.

For future work, we will allow the system to categorise the text and to generate a title for a non-title text. Also, we will enable the user to enter text by entering a URL of a web page and to send the summary by email. We will improve the system performance by summarising the generated summary if the user chose a different size on the same text.

References

1. Abbasi-ghalehtaki, R., Khotanlou, H., Esmaeilpour, M.: Fuzzy evolutionary cellular learning automata model for text summarization. Swarm Evol. Comput. **30**, 11–26 (2016)
2. Al-Abdallah, R.Z., Al-Taani, A.T.: Arabic single-document text summarization using particle swarm optimization algorithm. Proc. Comput. Sci. **117**, 30–37 (2017)
3. Al Qassem, L.M., Wang, D., Al Mahmoud, Z., Barada, H., Al-Rubaie, A., Almoosa, N.I.: Automatic Arabic summarization: a survey of methodologies and systems. Proc. Comput. Sci. **117**, 10–18 (2017)
4. Al-Saleh, A.B., Menai, M.E.B.: Automatic arabic text summarization: a survey. Artif. Intell. Rev. **45**(2), 203–234 (2016)

5. Alami, N., El Adlouni, Y., En-nahnahi, N., Meknassi, M.: Using statistical and semantic analysis for Arabic text summarization. In: Noreddine, G., Kacprzyk, J. (eds.) ITCS 2017. AISC, vol. 640, pp. 35–50. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-64719-7_4
6. Alami, N., Meknassi, M., Ouatik, S.A., Ennahnahi, N.: Arabic text summarization based on graph theory. In: 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), pp. 1–8. IEEE (2015)
7. Arabic Stop Words. Sourceforge (2018). http://arabicstopwords.sourceforge.net/
8. Bui, D.D.A., Del Fiol, G., Hurdle, J.F., Jonnalagadda, S.: Extractive text summarization system to aid data extraction from full text in systematic review development. J. Biomed. Inform. **64**, 265–272 (2016)
9. Douzidia, F.S., Lapalme, G.: Lakhas, an Arabic summarization system. In: Proceedings of DUC 2004 (2004)
10. El-Haj, M., Kruschwitz, U., Fox, C.: Using mechanical Turk to create a corpus of Arabic summaries (2010)
11. Evans, D.: Identifying Similarity in Text: Multi-Lingual Analysis for Summarization. Columbia University, New York (2005)
12. Fang, C., Mu, D., Deng, Z., Wu, Z.: Word-sentence co-ranking for automatic extractive text summarization. Expert Syst. Appl. **72**, 189–195 (2017)
13. Hovy, E.: Text summarization. In: The Oxford Handbook of Computational Linguistics, 2nd edn (2003)
14. Jaradat, Y.A., Al-Taani, A.T.: Hybrid-based Arabic single-document text summarization approach using genetic algorithm. In: 2016 7th International Conference on Information and Communication Systems (ICICS), pp. 85–91. IEEE (2016)
15. Palmer, D.D.: Tokenisation and sentence segmentation. In: Handbook of Natural Language Processing, pp. 11–35 (2000)
16. Reddy, P.V., Vardhan, B.V., Govardhan, A.: Corpus based extractive document summarization for Indic script. In: 2011 International Conference on Asian Language Processing (IALP), pp. 154–157. IEEE (2011)
17. RxNLP/ROUGE-2.0. GitHub (2018). https://github.com/RxNLP/ROUGE-2.0/tree/master/versions
18. Wu, Z., et al.: A topic modeling based approach to novel document automatic summarization. Expert Syst. Appl. **84**, 12–23 (2017)