RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Extractive Arabic Text Summarization Using PageRank and Word Embedding

Ghadir Alselwi[1] · Tuğrul Taşcı[1]

Received: 28 July 2023 / Accepted: 20 February 2024 / Published online: 18 April 2024

Abstract

Research on graph-based automatic text summarization for Arabic, the official language of 26 nations with over 200 million speakers, as well as other prevalent languages, has recently increased due to the ability of these approaches to handle linguistic peculiarities such as complex morphological linkages. The present paper proposes a graph-based extractive Arabic text summarization (GEATS) technique that employs word embedding and PageRank algorithms for feature extraction and sentence ordering. The efficiency of the GEATS approach versus the state-of-the-art methods is analyzed based on the quality of the produced summaries over the F-measure values. The findings indicated that it outperformed the nearest alternative by an advantage of over 7.5%.

Keywords Extractive text summarization · Arabic text summarization · PageRank algorithm · Word embedding · Farasa Stemmer

1 Introduction

The currently available digital data on several active online platforms, including websites, social media network apps, digital books, and scientific journals, have expanded significantly within the last decade and continue to increase tremendously. On the other hand, similar to all areas of life, instant access to information required in the current rapidly changing environment became an absolute necessity. Automatic text summarization (ATS), a diverse branch of natural language processing (NLP), provides reasonable solutions for the achievement of concise information based on the shortening of complex, long, and/or redundant textual documents (i.e., time-consuming).

A summary is a concise representation of the main points or ideas in a long text or speech, providing a condensed overview of the key content. There are two text summarization methods: extractive and abstractive. Extractive text summarization entails determination of important paragraphs or sentences in a document and combination of these elements without any modification. The aim is to extract relevant and significant parts of the original document. ATS, an active field of study, is obviously a difficult, computationally expensive, and time-consuming process due to the obvious need for human comprehension and linguistic skills. However, since the 1950s, a significant number of studies have been conducted on ATS in several languages, predominantly in English.

Numerous languages are spoken around the world that belong to different language families with various distinct dialects. Certain languages are spoken in large geographies and by populous nations. These include Arabic, Chinese (Mandarin), English, French, Russian, and Spanish (Castilian), which are the official languages of the United Nations (UN) and the mother tongue or second language of about 45 percent of the global population.

Due to the growing demand, the number of ATS tools that target different languages including Arabic, has increased. Although the popularity of Arabic text summarization (ArTS) has increased in recent years, the quality of existing tools requires improvement. Arabic is among the most commonly spoken languages in the world. More than 200 million people speak Arabic as a first language [1], and it is the official language in 26 nations. Arabic, a crucial business instrument, particularly in the Middle East and North Africa,

✉ Ghadir Alselwi
abdulhakim.alselwi@ogr.sakarya.edu.tr

Tuğrul Taşcı
ttasci@sakarya.edu.tr

[1] Information Systems Engineering, Sakarya University, Esentepe Campus, 54187 Serdivan, Sakarya, Turkey

does not only preserve the comprehensive cultural heritage of the Arab world but also serves as the liturgical language of Islam with more than 1.5 billion believers.

Several text summarization approaches and methods have been proposed in the literature, including Hidden Markov Model, Binary classifier, TextRank, graph-based methods, LexRank, Bayesian method, support vector machine (SVM), logistic regression model, decision trees, term frequency-inverse document frequency (TF-IDF), deep learning (DL), maximum marginal fitness algorithm, and clustering [2]. Several studies employed various methods including statistics, semantics [3, 4], machine learning (ML) [5], meta-heuristics [6–9], hybrid [10], deep learning [11, 12] and graph-based approaches [13, 14] in ArTS.

Quite a few studies focused on ArTS in recent years. Imam et al. [15] employed the Analogic Summarization System for Arabic Documents (OSSAD) a user-centered summarization system. Al-Taani and Al-Omour [4] addressed the semantic associations between the sentences with the graph-based short path algorithm (SPA). Jaradat and Al-Taani [9] applied a hybrid approach that investigated the impact of a scoring system based on the combination of informative and semantic scores to address the negligence of semantic associations between the sentences and the accuracy of the summary. Al-Abdullah and Al-Taani [6] employed the Particle Swarm Optimization (PSO) algorithm to achieve the best summary of a document by combining informative and semantic scores. Al-Radaideh and Bataineh [7] employed a hybrid approach that included statistical properties, semantic similarities, and genetic algorithm (GA) and claimed that the approach led to the generation of better summaries in terms of recall, precision, and F-measure scores. Al-Abdullah and Al-Taani [8] utilized FireFly (FF) algorithm that combined the informative and semantic scores to obtain better results. Qaroush et al. [3] proposed a generic extractive single document (SD) summarizing technique that included two strategies based on the score and machine learning. Arabic language NLP studies became popular in recent years. However, in the literature, it could be observed that these studies are still in early stages and require further research due to the lack of Arabic NLP resources [14].

Yet another and relatively novel ArTS category is the graph-based approaches. Graphs could be used to assist in the determination and reduction of the issues associated with the Arabic language due to their capacity to organize large and complex structures in standard forms. The field literature indicated that ArTS is still in its early stages and requires further research due to the lack of sufficient Arabic NLP resources. The present study aimed to propose a graph-based method for extractive summarization of Arabic texts that exist in a few available datasets. Due to the specific pre-processing stage, the proposed graph-based extractive Arabic text summarization (GEATS) method could introduce a coherent and complete summarization framework.

Despite extensive research on ATS, the particular linguistic properties of the Arabic language pose certain problems. Even after the latest advances, there is still room for improvement in the quality of summaries generated by current methods by focusing on distinct aspects. The present paper aimed to present a new approach constructed on well-known methods such as Google PageRank, cosine similarity, and word embedding. However, the novelty of GEATS is its meticulous adaptation of these methods to fit the subtleties of Arabic text, including a distinct pre-processing stage and extensive data flow.

The remainder of the paper is divided into the following sections: Text summarization-related works are examined in Sect. 2. Section 3 introduces the proposed method's data flow process and its operational logic, including the pursued pre-processing, feature extraction, and graph construction and weighting tasks. The findings and results are discussed in Sect. 4, revealing the selection of the best-fitted parameters and auxiliary methods, as well as the comparison against the state-of-the-art methods. The conclusion and remarks for the future are included in Sect. 5.

2 Literature Review

The increase in Arabic language applications and speakers has led to the significance of the field of Arabic NLP. Despite the efforts, the vast domain of Arabic NLP is still underexplored. Various techniques have been used to develop ATS, particularly in the Arabic language. Studies conducted between 2012 and 2023 explored techniques such as compression rate (CR), multi-document or single document, and various datasets and F-measures as present in Table 1 popular Arabic datasets such as EASC led to diverse achievements, typically represented by F-measures based on CR. The non-exhaustive content was discussed with a broader perspective, where methodologies and algorithms were investigated under five categories: statistics, graphs, meta-heuristics, machine learning, and deep learning.

Statistical text summarization methods employ various statistical properties such as word frequency, term weights, position in a sentence, semantic similarity, named entities, and co-occurrence patterns. These properties help the extraction of important summarization data. Statistical text summarization approaches could be employed without inherent memory and CPU problems [28]. Bialy et al. [29] proposed a statistical extraction method for Arabic SD summarization with three stages: pre-processing, sentence scoring, and summary generation. They compared their findings with summaries generated by human expert and determined that their findings were superior. Elayeb et al. [30] developed

Table 1 Recent ATS studies in Arabic language

Ref	Year	Corpus	Method	F (%)	CR (%)
[15]	2013	EASC	Ontology-based	49.8	40
[16]	2014	EASC	Clustering (K-Means algorithm)	60	–
[4]	2014	EASC	Graph-Based (Short path algorithm)	48.6	40
[17]	2015	ATE	Lexical Cohesion & Text Entailment Relation	69.98	–
[9]	2016	EASC	Hybrid-Based	54.76	40
[6]	2017	EASC	Particle swarm optimization	55.32	40
[7]	2018	EASC	Genetic algorithm	60.5	40
[8]	2019	EASC	Firefly algorithm	57.52	40
[18]	2020	EASC	Unsupervised Score-Based (Clustering, Word2vec)	64.4	30
[13]	2020	EASC	Graph-Based	76.37	–
[19]	2020	EASC	Modified PR algorithm	67.99	–
[3]	2021	EASC	Statistical and semantic features	64.3	50
[20]	2021	EASC	Documents clustering, unsupervised neural networks	20.68	40
[21]	2021	EASC	Genetic algorithm	41	–
[22]	2021	EASC	Knapsack balancing of effective retention	56.14	-
[23]	2021	EASC	ArDBertSum, DistilBERT model	49	–
[24]	2022	EASC	Statistical and Graph-Based	55.36	45
[25]	2022	AHS	Deep Learning (Seq2Seq model)	51.49	–
		AMN		44.28	
[26]	2022	EASC	Hybrid-Based	49.32	–
[27]	2023	EASC	Textual Graph	61.7	20

ATE, AHS, and AMN stand for Arabic textual entailment, Arabic headline summary, and Arabic Mogalad Ndeef, respectively

an extraction technique for Arabic SD summarization based on analogical proportions and implementation of two methods based on the presence of keywords in the document or summary and analysis of the frequency of these keywords. They employed ANT and a short version of Essex Arabic Summaries Corpus (EASC) test set to compare the two methods with the techniques based on Luhn, TextRank, LexRank, and LSA algorithms. The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) and Bilingual Evaluation Understudy (BLEU) were employed to compare these two methods. Promising results were reported when compared to the three other methodologies that employed the same datasets.

Metaheuristic approaches are effective tools in complex problems that are difficult or impossible to resolve optimally in polynomial time. These strategies employ evolutionary concepts based on biological evolution and natural events, and offer effective solutions. Metaheuristic algorithms are particularly beneficial in text summarization, since they are scalable, fast, and could tackle various problem domains. These techniques employ iterative procedures to investigate and improve alternative solutions over time, often relying on population-based strategies. PSO [6] was proposed for single-document ArTS and analyzed with the EASC dataset and the ROUGE measure. The findings were encouraging when compared to the current techniques that employed GAs [9] and Harmony Search algorithm [21]. Al-Radaideh et al. [7] suggested the SD ArTS technique for the EASC dataset, which combined GAs, statistical properties, and domain expertise to obtain the final summary. Al-Abdallah and Al-Taani [8] proposed an extraction method based on the FF algorithm that employed informative and semantic scores to determine the optimal sub-path among the candidate paths on the graph. The FF algorithm outperformed evolution-based approaches, Harmony Search, and GA recall, precision, and F-measure results. Alqaisi et al. [31] proposed an extraction technique for multi-document ArTS based on clustering and multi-objective optimization and employed the TAC 2011 and DUC 2002 datasets in the analysis. Hybrid text summarization approaches employ several methods and strategies to improve the summaries and produce better results when compared to the current systems. These methods include statistical properties, semantic similarity metrics, and genetic algorithms (GAs). Statistical properties such as word frequency and sentence position are employed to extract significant textual data, while semantic similarities determine the degrees of association between the text components. GAs optimize summaries based on evolving solutions produced by fitness analysis. Hybrid strategies combine quantitative and qualitative components, leading to a more thorough content identification. Semantic similarity captures the underlying meaning and context, while statistical properties determine the significant patterns. GAs provide a powerful optimization framework to efficiently search for and reduce the number of potential summaries. The GA component improves summarization with genetic operators and the measurement of individual fitness based on recall, precision, and F-measure [7].

Hybrid text summarization approaches improve the summary based on statistical properties, semantic similarity metrics, and graph-based analysis. These methods aim to produce more accurate and thorough summaries by combining the above-mentioned components, improving the F-measure, recall, and precision. Fadel and Esmer [10] proposed a hybrid approach that combined the abstraction and extraction techniques to achieve an informative and cohesive summary of a long text. Qaroush et al. [3] introduced an extractive Arabic SD summarization method based on supervised ML techniques and a mix of statistical and semantic data, and claimed better outcomes when compared to earlier techniques.

Machine learning has been increasingly employed in textual summarization to identify data patterns and associations, while reflecting linguistic structure and complexity. These models could be trained with various datasets available in multiple resources; thus, could be applied in various fields. Deep learning architectures such as RNNs or transformers excel in the recognition of semantic and contextual data, producing clear and relevant summaries. They maintain coherence and flow during the consideration of the context of larger texts. Studies conducted by Molham and Said [12] and Wazery et al. [25] introduced robust deep learning abstractive summarization systems for Arabic text, outperforming previous methods. These models demonstrated contextual awareness about the larger text context, while sustaining coherence and maintaining flow. Although those techniques are excellent, the method presented in the present paper adopted a different approach based on a relatively smaller dataset. It prioritized the preservation of the essential data and to overcome language-specific challenges while consuming low computational power. Ellouze et al. [5] developed a novel ML approach to automatically analyze the overall responsiveness of ArTS based on linguistic properties and content scores such as ROUGE, MeMoG, and SIMetrix. Lamsiyah et al. [11] presented a deep learning method based on language embedding model for SD ArTS, and demonstrated its effectiveness when compared to eight previous techniques. Molham and Said [12] developed a dataset of Arabic summaries based on two abstraction Arabic literature summarization models. Suleiman and Awajan [32] proposed an RNN method for abstractive summarization of Arabic texts that employed two levels of hidden states at the encoder and a single stage at the decoder. Empirical findings revealed that the suggested method performed well with certain types of ROUGE. Machine learning techniques are effective in the generation of succinct and detailed summaries in various applications due to their ability to learn from the data, adapt to different domains, understand context, improve continuously, scale effectively, and customization capabilities.

Graph-based approaches focus on semantic associations between sentences in a document, where nodes represent sentences and edges represent semantic associations. Al-Taani and Al-Omour [4] analyzed the impact of primary elements such as word stems, words, and n-grams on the performance of an extractive graph-based ArTS technique. The proposed method employed n-grams in summarization, was tested with the EASC dataset, and the authors claimed that it outperformed earlier approaches. Elbarougy et al. [33] studied the effect of stopwords removal during pre-processing on the performance of a graph-based ArTS technique. They conducted two experiments, where one involved the generation of a summary without discarding stopwords and the stopwords were completely eliminated in the other. The findings demonstrated that the removal of the stop-words improved summarization performance. Elbarougy et al. [19] investigated the impacts of various morphological analyzers on the overall performance of a graph-based method with the EASC dataset. In another study conducted by Elbarougy et al. [13], the impact of various morphological analyzers on graph-based method performance was analyzed with the EASC dataset. They reported that the middle analyzer, BAMA, outperformed the others and produced superior results.

Statistical approaches require less CPU power and memory [34]; however, they might exclude key sentences or include irrelevant ones. ML approaches require large manually labeled training data [35] to produce a reasonable summary. Metaheuristic algorithms have the advantage of computing ideal weights but these require significant computation time, cost, and iterations [36]. Graphical methods improve coherence, identify duplicate data, and are independent of language and domain [35]. Researchers have accepted these as a powerful option for the organization of complex structures with standard and formal methods that employ graphs to overcome lingual challenges in Arabic. Graphical methods are language and domain-independent; thus, they are a powerful choice to overcome language barriers.

3 GEATS Approach for Arabic Text Summarization

Due to the complexity of the Arabic language and the paucity of previous studies, Arabic summarization continues to perform poorly. Several ArTS studies have employed graph-based algorithms such as PR algorithm, and TF-IDF in text representation and feature extraction. For feature extraction, the present study employed the PR and WE methods on

Word2Vec. Furthermore, the present study sought to employ three algorithms, namely PR, LexRank, and TextRank, to achieve the best performance.

The present study aimed to develop a model to generate automatic ArTS based on extractive approaches that could be applied to several domains with excellent performance. To achieve this objective, we needed to determine certain parameters such as an adequate list of stopwords and the proper data corpus, the most adequate number of preprocessing steps, the most applicable stemming technique, the best primary elements, the most relevant extraction properties, an adequate dataset to test the system, a summary reconstruction method, and the proper approach to analyze the summary.

3.1 Graph-Related Concepts and Methods

A graph G represents the text input. A directed graph $G = (V, E)$ is the graph G for document D, where V is the collection of nodes and E is the set of edges [37]. V and E are the two primary components of the graph. In other words, G is a weighted directed network the nodes of which represent D sentences and the edge weights of which indicate sentence similarity. $G(V, E)$ is a mathematical structure that represents the pairwise correlations between the items. Edges reflect the nature of the correlation between the two vertices, while the vertices represent the primary component of the depicted system. The employment of a graph-based model to solve a textual summarization problem should address three primary issues: (1) fundamental components of the application such as words, phrases, sentences, or paragraphs (2) the type of correlation between the nodes to determine the edge weights such as cosine similarity or overlapping phrases, etc., (3) and an algorithm to rank the vertices of the graph such as LexRank [38], TextRank [39], or PageRank [40].

TextRank is a graph-based, single-document ranking method adopted by the Google PageRank algorithm [40]. The similarity between the phrases is represented with an edge weight in TextRank, an undirected linked graph. TextRank extracts the sentences and keywords. Then, the sentences are ranked based on their scores, where the highest-ranked sentences are selected for the summary. LexRank is a graph-based multi-document summarization method where all sentences are represented in a graph. Two sentences are associated when their similarities exceed a certain threshold. After the graph is plotted, the most central sentences are selected for the summary.

An essential property of a node in complex networks such as the World Wide Web (WWW) is its in-degree (out-degree), which reflects the number of inbound (outbound) links to the node [41]. The in-degree of a specific page could be an approximation of the significance or quality of that page [42]. The PageRank algorithm [42] expanded on this concept by not considering the incoming links from all pages equally but normalizing the links based on their significance and the number of outbound connections from nearby pages. Thus, the PR could be a superior measure of relevance since it integrates the product's visibility and authority based on the number of citations and the reputation of the citing publications [42]. $PR(A)$ is determined by a simple iterative method that corresponds to the primary eigenvector of the web's normalized link matrix [42]. The PR of the Web page A, as indicated by PR, is defined with Eq. (1).

$$\mathrm{PR}(A) = (1 - d) + d * \sum_{i} \frac{\mathrm{PR}(T_i)}{C(T_i)} \tag{1}$$

where $\mathrm{PR}(T_i)$ is the PR of page T_i that is linked to page A, $C(T_i)$ is the number of outbound links on page T_i, and d is a damping factor that could be set between [0, 1].

The PR of A is recursively determined with the PR of the pages that link to page A, as shown in Eq. (1). The PR of the page T_i is always weighted by the number of outbound connections $C(T_i)$ inside the algorithm, leading to a lower PR transmitted from the page T_i to the receiving page A. It was also anticipated that each new inbound link to the recipient page A would always boost A's PR.

3.2 GEATS Approach

The proposed approach is discussed in this section. The GEATS flowchart, which includes three primary steps, is presented in Fig. 1. The first stage entails the extraction of the text from a document, followed by pre-processing procedures such as normalization, removal of the stopwords, and stemming. The desired properties are retrieved in the second stage, and the document is plotted on a graph. Finally, the PR method is employed to generate the summary in the third stage, after which the performance is analyzed and the findings are presented.

The Arabic language was classified as a language with wealthy and complex morphological and syntactic flexibility [43]. Thus, tackling Arabic documents without a pre-processing phase in data retrieval would most likely lead to a more challenging text and inaccurate findings. Therefore, the proposed GEATS method involves certain pre-processing steps that target words, sentences, and the whole document of analysis.

The proposed method initially imports the documents in the external dataset, i.e., the well-known EASC corpus. The GEATS summarization includes a normalization phase for each sentence in each document, which entails the transformation of the text into another format to improve consistency. Normalization plays a key role in the quality of the final summary since it involves the removal of repetitive phrases, duplicated spaces, etc. Normalization includes the following steps: (1) removal of diacritics or Tashkeel which are applied

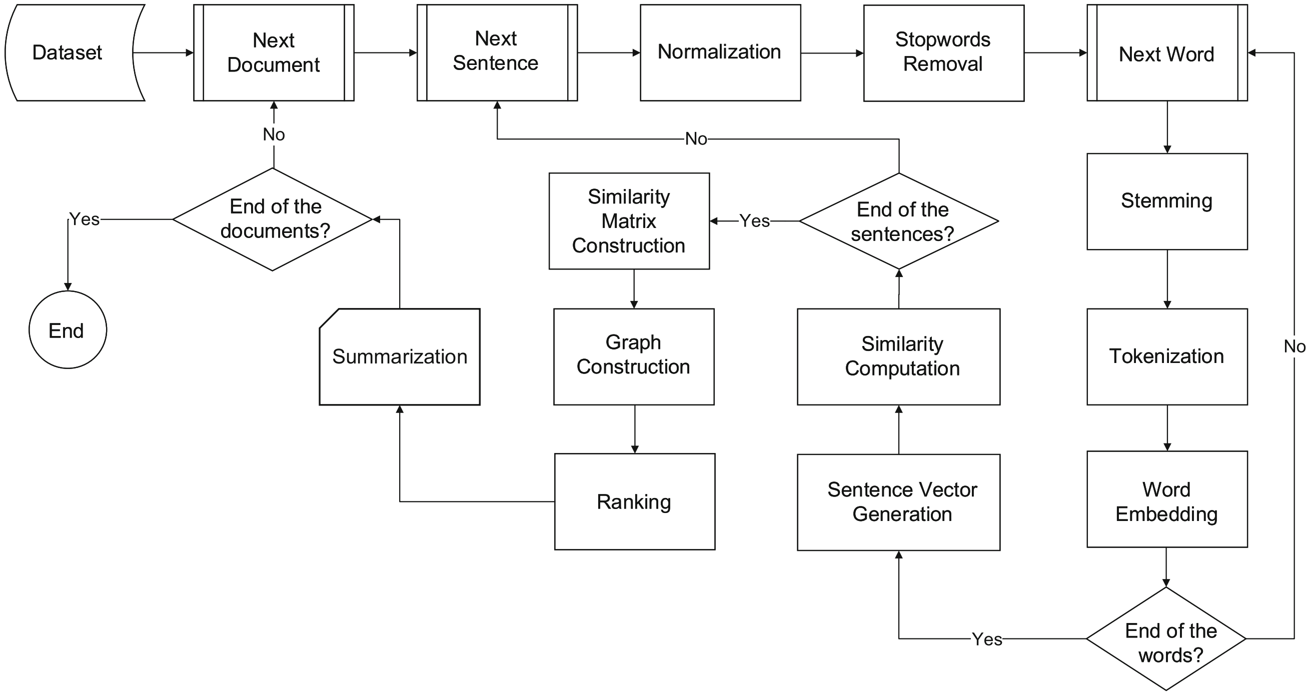

Fig. 1 GEATS flowchart

to the Arabic letter "ر" such as "رَ" Fatha, "رً" Tanwin Fath, "رُ" Damma, etc, (2) removal of punctuations and links, duplicate spaces, numerous full stops, and (3) check and replacement of ALEF Styles "إ أ آ آ ا" in every word with a unique form "ا".

3.2.1 Pre-processing Phase

Arabic includes distinctive notations known as diacritics. They are used to assist Arabic readers to correctly pronounce these Arabic words. Diacritics are assigned based on the Arabic grammar rules. The change in the location of a word in the sentence leads to various diacritics and meanings. Although diacritics are particularly important in Arabic texts, they could be omitted from summarization to achieve a more representative summary. The diacritics available in Arabic are presented in Table 2.

Table 2 Diacritics for the Arabic letter "ر"

Name	Shape	Name	Shape
Fatha	رَ	Shadda and Tanwin Dam	رٌ
Dama	رُ	Shadda and Tanwin Kasr	رٍ
Kasra	رِ	Shadda and Tanwin Fath	رً
Tanwin Fath	رً	Shadda Dam	رُّ
Tanwin Dam	رٌ	Shadda Fath	رَّ
Sukun	رْ	Shadda Kasr	رِّ
Madah	~	Shadda and Sukun	رّْ
Shadda	رّ	Tanwin Kasr	رٍ

An example of deleting diacritics from sentences is shown in this text since the initial text "نَزَل عَلَيْكَ الْكِتَابَ بِالْحَقِ...", and the result after removing diacritics is presented in this sentence "نزل عليك الكتاب بالحق...".

Arabic, similar to other languages, requires several marks to organize the texts and provide readers with the proper meaning of sentences. At this pre-processing step, the input text is cleaned up of the characters presented in Fig. 2, which could be considered punctuation marks in Arabic manuscripts. An example of punctuation's removal process is presented in this text since the initial text is "(تحريف بشكل عام)", and the result after punctuation removal is presented in this sentence "تحريف بشكل عام".

ALEF is the first character in the Arabic alphabet. ALEF could be written in multiple forms or shapes such as "إ أ آ آ ا" based on its position in the word. The system converts each ALEF in the text to "ا", ensuring a uniform format, which helps stemming. The unification operation conducted on ALEF in each word is presented in Table 3.

Elimination of all stopwords is an essential step in the normalization phase of GEATS summarization. Stopwords typically serve as connectors that improve the semantics

<table>
<tr><td>< ></td><td>'</td><td>{ }</td><td>~</td><td>¦</td><td>+</td><td>#</td><td>@</td><td>÷</td><td>» «</td></tr>
<tr><td>`</td><td>|</td><td>…</td><td>“ ”</td><td>–</td><td>.</td><td>[ ]</td><td>?</td><td>!</td><td>;</td></tr>
<tr><td>؛</td><td>‘,</td><td>" "</td><td>/</td><td>:</td><td>"</td><td>\</td><td>×</td><td>] [</td><td>‹ ›</td></tr>
<tr><td>_</td><td>( )</td><td>*</td><td>&</td><td>,</td><td>%</td><td>=</td><td>$</td><td>^</td><td>..</td></tr>
</table>

Fig. 2 Eliminated punctuation marks

Table 3 In-word ALEF use unification

Original Text	After unification
تتباين رؤية الإسلام عن الرؤية اليهودية في مسألة التوراة، إذ يتّفق الفريقان أن التوراة من عند الله أنزلها على موسى كما توضّح الآية ٣ من سورة آل عمران (نَزَل عَلَيْكَ الْكِتَابَ بِالْحَقِ مُصَدِقَاً لِما بَيْنَ يَدَيْهِ وَأَنزَلَ التَورَاةَ وَالإِنجِيلَ.. والآية ٥٣ من سورة البقرة (وَإِذْ آتَيْنَا مُوسَى الْكِتَابَ وَالْفُرْقَانَ لَعَلَكمْ تَهْتَدُونَ). فالتوراة من عند الله ولكن يعتقد المسلمون بأن توراة اليوم طرأ عليها زيادة ونقصان (تحريف بشكل عام) مقارنة بالتوراة المنزّلة على موسى.	تتباين رؤية الاسلام عن الرؤية اليهودية في مسالة التوراة، اذ يتّفق الفريقان ان التوراة من عند الله انزلها على موسى كما توضّح الاية ٣ من سورة ال عمران (نَزَل عَلَيْكَ الْكِتَابَ بِالْحَقِ مُصَدِقَاً لِما بَيْنَ يَدَيْهِ وَانزَلَ التَورَاةَ وَالاِنجِيلَ..) والاية ٥٣ من سورة البقرة (وَاِذْ اتَيْنَا مُوسَى الْكِتَابَ وَالْفُرْقَانَ لَعَلَكمْ تَهْتَدُونَ). فالتوراة من عند الله ولكن يعتقد المسلمون بان توراة اليوم طرا عليها زيادة ونقصان (تحريف بشكل عام) مقارنة بالتوراة المنزّلة على موسى.

of a phrase. Stopwords could be categorized as adverbs, prepositions, relative, verbal, conditional or basic pronouns, measurement units, referrals or determiners, and transformers (verbs, letters), etc [44]. In Arabic, these words are present throughout the text in "في، من، إلى، على" forms. In general, the removal of the stopwords improves data retrieval efficiency since common words tend to reduce frequency differences and the length of the document, which affects weighing [44].

In fact, there is no particular or publicly accepted list of stopwords in Arabic language. The words differ based on the author and/or text and context. In the present paper, the Arabic stopwords list was based on the Natural Language Toolkit (NLTK[1]). The removal operation is quite important for the elimination of all stopwords.

Table 4 Stopwords removal

Original Text	After Stopwords Removal
الفضة من المعادن الكريمة ابيض اللون، وهو معدن ثمين معروف منذ القدم حيث عرفه قدماء المصريين والعرب والصينيون واستخدموه في صناعة الحلي وفي الطب والوقاية من الامراض. تستخدم في النقود والحلي تماماً كالذهب الا أنها اقل قيمة.	الفضة المعادن الكريمة أبيض اللون، معدن ثمين معروف القدم عرفه قدماء المصريين والعرب والصينيون واستخدموه صناعة الحلي وفي الطب والوقاية الأمراض . تستخدم النقود والحلي تماماً كالذهب أنها قيمة .

An example of stopwords removal is presented in Table 4, including the original text and the text after the removal of the stopwords.

The technique employed to reduce the words to their origins or roots is known as stemming. Roots or primal forms of the words are obtained by stemming any applied affixes. The goal is to obtain the original term or the root of the word, to improve the associational weighing of the sentences, improving the quality of the summarization.

For instance, stemming the Arabic word ‘writing’ "كتابة" provides the source ‘write’ "كتب". The word ‘writer’ "كاتب" would also provide the same root. After stemming, the resulting words could be utilized in various applications such as compression, spell-checking, and textual search. Thus, Farasa stemmer [45] was employed to determine the root of every word in the sentences as illustrated in Table 5. This process minimized the number of unique words in the document to achieve better term frequency computations in vector representation.

Another procedure adopted in GEATS summarization was tokenization, also known as segmentation. It entails the reduction of all texts to a lower number of elements. During this process, document text is separated into paragraphs, sentences, and words respectively. For instance, the sentence [لا تدعى الطفل يخرج يده أو رأسه خارج نافذة السيارة.] is tokenized into individual words as follows: ["لا", "تدعى", "الطفل", "يخرج", "يده", "أو", "رأسه", "خارج", "نافذة",

[1] https://www.nltk.org/.

Table 5 Stemming example

Original Text	After Stemming
الفضة من المعادن الكريمة ابيض اللون، وهو معدن ثمين معروف منذ القدم حيث عرفه قدماء المصريين والعرب والصينيون واستخدموه في صناعة الحلي وفي الطب والوقاية من الامراض. تستخدم في النقود والحلي تماماً كالذهب الا انها اقل قيمة.	فضة من معدن كريم أبيض لون ، هو معدن ثمين معروف منذ قدم حيث عرف قديم مصري عرب صيني استخدم في صناعة حلي في طب وقاية من مرض . استخدم في نقد حلي تمام ذهب إلا أن أقل قيمة .

"السيارة."]. The output of the tokenization operation is actually the input for the stemming operation.

3.2.2 Feature Extraction Phase

The feature extraction step, a cornerstone of the extractive summarization, is critical for the determination of the essence of the text and the identification of key summary data. This phase entails the collection of two types of significant data: the vector representation of the root words, known as terms, and the measure of similarities between the unique sentences.

To achieve the required representation, a full vocabulary was constructed with the training dataset after the input corpus was pre-processed. Word2Vec, a known NLP approach, was employed to generate an embedding vector for each phrase. This stage not only identified the semantic associations between the words but also improved the model's capacity to recognize contextual subtleties of the language.

Word2Vec employs a high-dimensional space where each word in the lexicon is embedded in a separate vector. The model could capture semantic associations and meanings based on the placement of these vectors, which are based on the contextual similarities between the words. Thus, words with similar meanings are represented by the closer vectors in the constructed space.

The cosine similarity, a pivotal metric in gauging the similarities between texts, is based on the calculation of the cosine angle between two vectors, each of which encapsulates the essence of a particular text in the vector. This metric yields a numerical measure that ranges between 0 and 1, where 1 identifies the identical vectors. The comparison of the sentence vectors in each document leads to the selection of the sentence with the highest similarity to represent the concept in the summary.

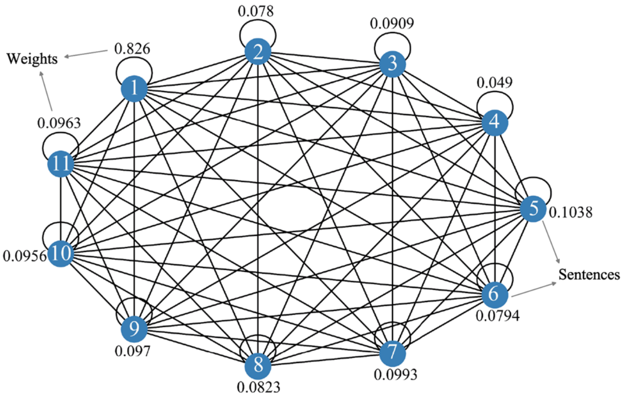

Fig. 3 The post-ranking graph-representation of a document

The following steps were conducted to calculate the similarity between two sentences (S_i, S_j) in the same document: (1) Acquisition of the vectors for both sentences; (2) Identification of the similar or associated words in the n-dimensional list, and (3) Iterative application of related words in the n-dimensional list; and (3) Iteratively apply the cosine similarity, calculated with Eq. (2), to the list elements.

$$CosineSimilarity(S_i, S_j) = \frac{\sum_1^n Word2Vec(S_i) * Word2Vec(S_j)}{\sqrt{\sum_1^n Word2Vec(S_i)^2 * \sum_1^n Word2Vec(S_j)^2}} \quad (2)$$

3.2.3 Graph Construction and Weighting

In this stage, each sentence is represented as a node, and each edge reflects the previously calculated cosine similarity between the two nodes or sentences. Thus, the document-level representation yields a fully associated graph, as presented in Fig. 3, where the nodes are the active document sentences and the edges between any two nodes represent the weights associated with the cosine similarity of the two.

After the graph was plotted and weighed, each document was ranked to achieve the final document summary. The PageRank method was employed in this process. A rating score was calculated for each node and the algorithm output and the nodes were ranked based on these scores. The best 'n' sentences were selected based on the highest number of cut-off sentences in the summary. The 'n' is a global parameter called compression ratio or CR that represents a specific ratio of sentences in the original text. After the determination of the summary, the sentences were re-ordered for best representation and to improve precision.

4 Findings and Analysis

No method could be considered the gold standard in text summarization. In other words, several summaries could be developed for each document depending on the human who developed the summary and the educational and technological background of that individual. The review of the publications on text summarization revealed that the human evaluators do not agree on a single summary for each paragraph. Thus, the analysis of text summarization is quite challenging.

4.1 Dataset (Corpus) and Analysis Metrics

The proposed methodology was analyzed with the EASC corpus generated by Mechanical Turk (Mturk) [46]. EASC includes 153 documents, each has five summaries developed by humans and collected from three resources: Wikipedia (106), Alwatan newspaper (34), and Alrai newspaper (13). The text documents in the EASC corpus could be classified into ten main categories: religion, education, science and technology, environment, finance, tourism, health, politics, sports, art, and music. There are summaries in the data set for each document which includes an average of 17 sentences.

An analysis was conducted to measure the quality of the summaries generated with the proposed GEATS approach versus the human-generated, or so-called ground-truth summaries. In the analysis, three well-known metrics, precision, recall, and F-measure [47], which have also been used for comparison in pattern recognition, data retrieval, and machine learning applications, were employed. Among these metrics, precision is a measure of quality, while recall measures quantity. F-measure is considered a type of equilibrator metric that entails the harmonic mean of precision and recall metrics. Recall, also known as sensitivity in binary classification, was computed in the present study by dividing the number of correct sentences by the expected outcomes (Eq. 3).

$$Recall = \frac{gram_{ref} \bigcap gram_{gen}}{gram_{ref}} \tag{3}$$

Precision, also known as confidence in data mining, was computed by dividing the number of correct sentences by the total number of generated outcomes (Eq. 4).

$$Precision = \frac{gram_{ref} \bigcap gram_{gen}}{gram_{gen}} \tag{4}$$

Valid for both Eqs. (3) and (4), the ground-truth or reference summary grams are represented by $grams_{ref}$ and the generated ones are depicted with $grams_{gen}$.

The employment of only recall or precision is inadequate, but the F-measure calculated with the Eq. (5) balances these two parameters and considers the overall performance of the estimator rather than only accuracy.

$$F - measure = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{5}$$

The ROUGE metrics quantify the number of overlapping elements across the computer-generated summary and the ideal summaries developed by humans, including n-grams, word sequences, and word pairs [48]. ROUGE-N, given in Eq. (6), is a generic calculation of the ROUGE metric. “N” is the number of overlapping contiguous elements included in the computation.

$$\text{ROUGE-}N = \frac{\sum_{s\in Ref\,Sums}\sum_{N-gram\in S} Count_{match(N-gram)}}{\sum_{s\in Ref\,Sums}\sum_{N-gram\in S} Count(N-gram)} \tag{6}$$

where N is the total size of the $N-gram, count_{match(N-gram)}$ is the highest number of grams in the computed and human(ground-truth) summaries, and count over $N - gram$ is the total number of $n - grams$ in the human summary.

In the present study, the ROUGE-1 and ROUGE-2 metrics were preferred where the comparison was conducted with the unigrams and bigrams. One of the other metrics employed in the study was the BLEU [49] score. The BLEU score, given in Eq. (7), is a modified precision metric, and a comprehensive measure of the linguistic similarity between the generated and reference summaries. The BLEU score provides the degree of similarity between 0 and 1, where 1 denotes identicality.

$$P_n = \frac{\sum_{C\in\{D\}}\sum_{n-G\in C} Count_{clip}(n-G)}{\sum_{C'\in\{D\}}\sum_{n-G'\in C'} Count(n-G')} \tag{7}$$

where $n - G$ and D depict $n - gram$ and $Candidates$. $Count(n - gram)$ refers to the number of candidate n-grams in the test set and $Count_{clip}(n - gram)$ represents the count of clipped n-grams for the candidate sentences. The n-gram matches were initially computed for each sentence. The clipped n-gram count for all candidate sentences were summed and divided by the number of candidate n-grams in the test corpus to obtain the BLEU score.

4.2 Implementation

In this study, five possible scenarios were implemented to demonstrate the efficiency of the proposed approach. In the feature extraction stage, the performances of the two prominent methods, word embedding and TF-IDF, were compared.

The Farasa[2] and Arabic Light Stemmer, two available alternative stemmers, were compared in the second scenario. In the third scenario, the performance of the GEATS method was investigated based on various CR levels which reflected the potential summary inconsistency. The fourth scenario entailed the analysis of the performances of the GEATS, LexRank, and TextRank methods based on the same CR parameter. The final scenario compared the proposed GEATS method with the state-of-the-art methods that claimed to produce adequate summaries for Arabic texts. All scenarios were run on an Intel(R) Core(TM) i7-4810MQ CPU @ 2.80GHz computer with 16 Gigabyte RAM and Python language, NLP-oriented GENSIM, and NLTK libraries. Graph-based representation of the document is presented in Fig. 3 after the PR algorithm was applied. Each graphical node represents a sentence with a unique rank or weight. CR parameter was set to 40% for this illustration to achieve a 4-sentence long summary that included the nodes representing the sentences 5, 7, 9, and 11, with the highest weights.

4.3 Scenario-A: Word Embedding and TF-IDF Methods as Feature Extractors

In Scenario-A, the efficiency of Word Embedding (WE) and TF-IDF feature extraction methods was analyzed to determine the optimal mathematical representation of the vocabulary with the GEATS approach. The BLEU, precision, recall, and F-measure determined with the above-mentioned methods are presented in Fig. 4. Although the TF-IDF and WE produced comparable findings in certain criteria, the latter regularly led to better average percentages, and it performed particularly well in ROUGE-1 and ROUGE-2 metrics. The BLEU metric slightly favored TF-IDF with 0.1 superiority over Word Embedding (WE). However, WE consistently exhibited robust overall performance across various metrics, solidifying its superiority in quantitative representation of the dataset sentences based on the comparison results.

Unlike the TF-IDF, which represents a single word with a single vector, the WE provides a distributed vector representation of a word in a fixed-dimensional semantic space [50]. Thus, each word is represented by a multidimensional dense vector, reflecting its meaning nuancedly and context-aware, suggesting that the WE method led to a more comprehensive and meaningful representation of the words in the documents; therefore, it was a better feature extraction method in GEATS.

4.4 Scenario-B: Farasa and Arabic Light Stemmer

Farasa is a widely utilized stemming tool and library in the determination of word roots. The Arabic Light Stemmer (ArLS), known as Tashaphyne,[3] is an alternative stemming tool in Arabic. In Scenario-B, these two stemmers were compared with the proposed GEATS method and the CR was 40% in the five ground-truth summaries (S1 to S5). The BLEU, ROUGE-1, and ROUGE-2 metrics were calculated for the competing stemming tools.

The findings presented in Table 6 clearly demonstrated that the Farasa stemmer produced better F-measures in ROUGE-2, indicating that could capture longer sequences based on words and phrases. On the other hand, the ArLS stemmer yielded slightly higher F-measures in ROUGE-1 and BLEU metrics, suggesting its superior performance in capturing single-words and phrases.

Influencing the selection of the most proper stemming tool based on the specific summarization goals, and linguistic characteristics of the Arabic language, comparative results play a pivotal role in the performance of the proposed GEATS method. Farasa is selected as the stemming method in line with the results achieved in Sceneario-B.

4.5 Scenario-C: The Performance of the GEATS Method Various CR Values

The CR parameter represents the ratio of the summary length to the main text length and plays a key role in the determination of the performance of a text summarization method. A relatively small CR indicates that the number of sentences in the summary is too few, leading to the presentation of inadequate data about the main text. This could lead to lack of consistency in the summary. When the specified CR is too high, this could lead to a long summary that includes redundant sentences, verbosity and low readability.

In the present study, the acceptable CR range was accepted as 30–40%. This was considered adequate since it provided a balance between a concise summary and inclusion of relevant data. The F-measure values, a common analytical metric, which were collected with the proposed GEATS method for ROUGE-1 and ROUGE-2 based on the variations in CR are presented in Table 7. The review of Table 7, and Fig. 5 would demonstrate that the best CR parameter was 40%, yielding the best summarization quality. Thus, CR was accepted as 40% in GEATS implementation.

4.6 Scenario-D: GEATS Method Versus LexRank and TextRank

LexRank and TextRank have been recognized and commonly used in recent NLP research. In scenario D, the performance of the proposed GEATS approach was compared with the two above-mentioned methods and the same CR value (i.e.,

[2] https://alt.qcri.org/farasa/.

[3] https://pypi.org/project/Tashaphyne/.

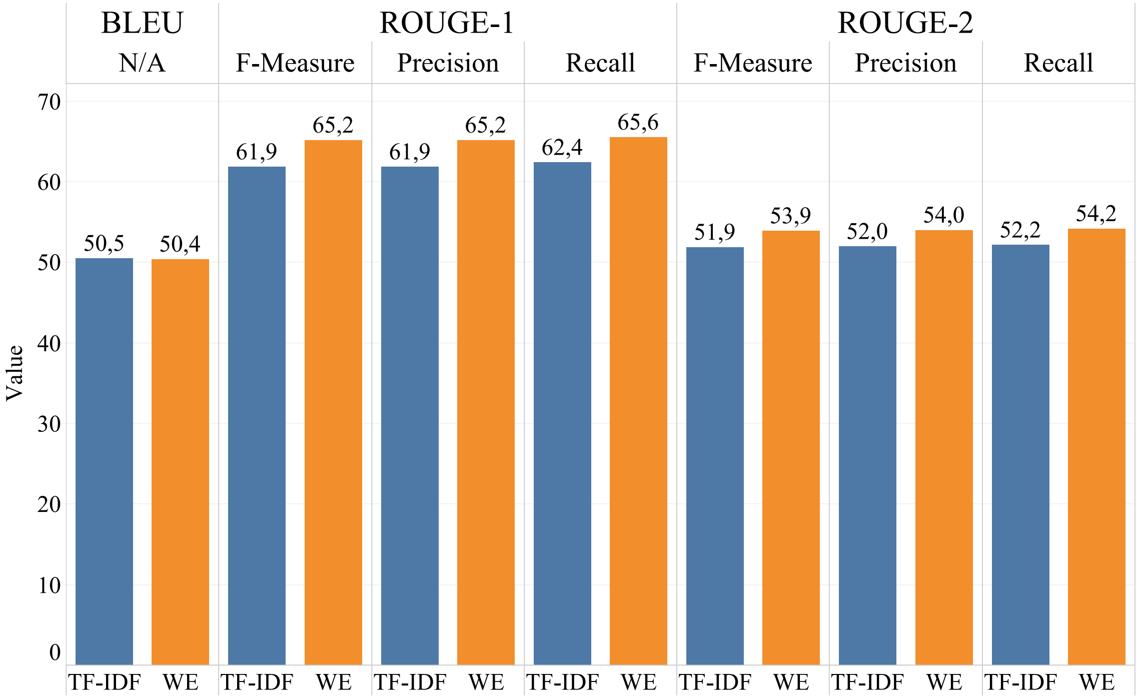

Fig. 4 The WE and TF-IDF methods as feature extractors

Table 6 The comparison of Farasa Stemmer and Arabic LS findings

Stemmer	Farasa			Arabic LS		
ROUGE-N	ROUGE-1	ROUGE-2	BLEU	ROUGE-1	ROUGE-2	BLEU
Metric	F	F		F	F	
Summaries						
S1	64.159	53.493	50.264	64.896	52.567	51.294
S2	65.679	54.128	50.569	66.144	53.695	51.539
S3	64.916	53.497	50.418	65.795	53.594	51.355
S4	65.841	54.508	50.408	65.141	53.267	51.343
S5	65.500	54.241	50.487	65.392	52.777	51.491
Mean	65.219	53.973	*50.429*	65.474	53.180	*51.404*

40%). The findings are presented in Table 8, where P, R, and F are precision, recall, and F-measure, respectively.

The comparative findings and precision, recall, and F-measure values demonstrated that the proposed GEATS method performed slightly better when compared to the LexRank method. This indicated that the GEATS method could produce more precise summaries, exhibit higher recall, and achieve better overall F-measure when compared to LexRank. When compared to the TextRank, GEATS method produced competitive results based on precision, recall, and F-measure, except the TextRank performed better in ROUGE-1 metric.

On the other hand, the GEATS method outperformed TextRank in the ROUGE-2 metric, which measures the overlap of adjacent word pairs (bigrams) in produced and reference summaries. Thus, the GEATS method identified coherent and semantic correlations between bigrams more effectively. While TextRank excelled in capturing individual word overlaps, the GEATS method was superior in capturing bigram overlaps, which is crucial for the preservation of contextual data and coherence in the summary. Thus, either TextRank or GEATS could exhibit better performance based on the specific analysis metric. The selection of one of the methods should be based on the summarization requirements and the significance of word overlap (ROUGE-1) versus bigram overlap (ROUGE-2). In summary, certain metrics indicated that TextRank was advantageous, the GEATS method excelled in capturing bigram overlaps, demonstrating that it could preserve contextual data and sentence coherence; thus, offering a significant alternative for summarization tasks with specific requirements. These findings suggested that the proposed GEATS method is a promising approach for

Table 7 ROUGE-1 and ROUGE-2 Precision, Recall and F-measure scores based on variable CR values

CR	ROUGE-1 Precision	Recall	F-measure	ROUGE-2 Precision	Recall	F-measure
10%	37.908	37.908	37.908	23.791	23.791	23.791
15%	47.843	47.843	47.843	29.019	29.019	29.019
20%	55.033	55.033	55.033	35.556	35.556	35.556
25%	58.316	58.193	58.201	42.462	42.437	42.389
30%	62.627	62.551	62.459	47.752	47.527	47.475
35%	65.855	66.086	65.745	52.301	52.338	52.165
40%	65.256	65.693	65.219	54.067	54.249	53.973
45%	62.645	63.521	62.584	52.289	52.698	52.285
50%	59.935	61.837	60.136	49.423	50.191	49.492

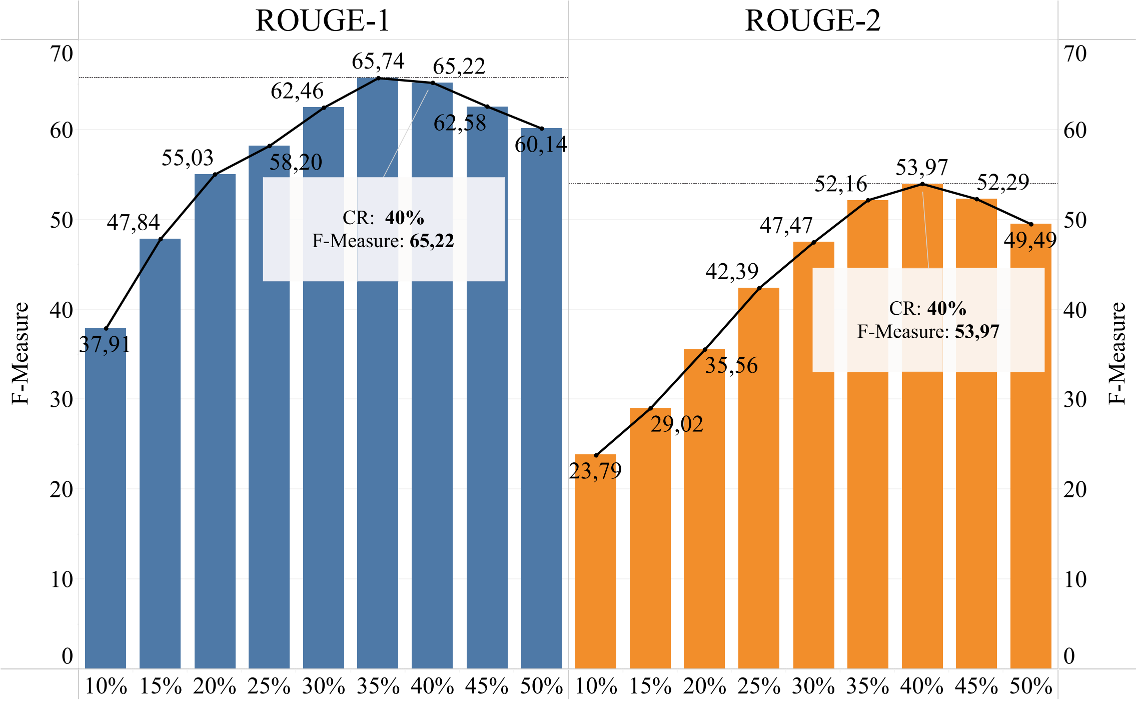

Fig. 5 ROUGE-1 and ROUGE-2 F-measurement values depending on CR change

Table 8 Comparison of the LexRank, TextRank, and (PageRank-based) GEATS Methods

Method	LexRank			TextRank			GEATS		
Metric	P	R	F	P	R	F	P	R	F
ROUGE-1	63.788	64.353	63.723	67.916	68.804	68.011	65.256	65.693	65.219
ROUGE-2	52.240	52.344	52.170	53.218	54.713	53.235	54.067	54.249	53.973

textual summarization, since it outperformed LexRank and exhibited comparable performance with TextRank in most cases.

4.7 Scenario-E: Performance Comparison of the GEATS Method and Other State-of-the-Art Methods

In the final scenario, the performance of the GEATS method was compared to six state-of-the-art approaches in text summarization. These approaches include ontological OSSAD

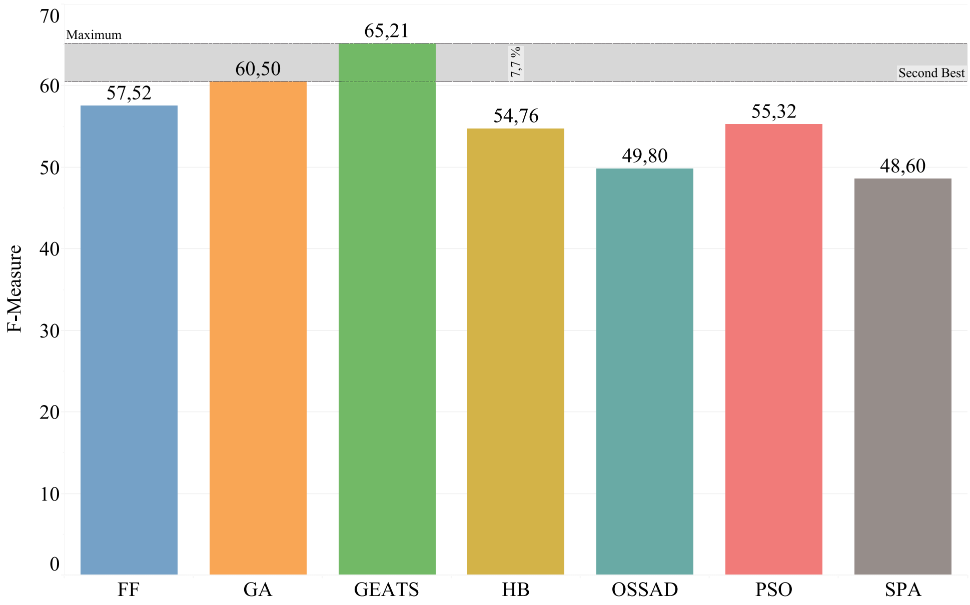

Fig. 6 GEATS versus the state-of-the-art methods

[15] method, the graphical SPA (Short Path Algorithm) approach [4], the hybrid HB approach [9], the PSO (Particle Swarm Optimization) algorithm [6], the GA (Genetic Algorithm) [7], and the FF (FireFly) algorithm [8].

The findings presented in Fig. 6 demonstrated that the proposed GEATS method outperformed the other reviewed methods based on the F-measure. The F-measure, commonly used to analyze the effectiveness of text summarization methods, takes both precision and recall into account, providing a balanced measure of summarization quality, indicating that the GEATS method could generate more accurate and comprehensive summaries when compared to the other approaches. The comparison revealed that the GEATS method could be a promising text summarization alternative and provide improved results when compared to the existing methods.

4.8 Execution Time and Memory Usage Considerations

In the detailed analysis of the execution time and memory usage for each scenario, notable differences emerge among the methods. In Scenario-A, the execution time for the WE method stands at 64.02 s, which is approximately 32% faster than the TF-IDF method's execution time of 96.01 s. Moreover, the memory requirement for WE is 1.08 GB, indicating a 20% reduction compared to the 1.35 GB memory usage by TF-IDF. Moving to Scenario-B, where Farasa and ArLS stemmers are employed, a substantial contrast is observed. Farasa, with a staggering stemming time of 5637.19 s, utilizes significantly less memory (0.65 GB) compared to ArLS, which has a substantially shorter stemming time of 216.89 s but requires more memory (1.38 GB). In Scenario-C, the variation in Compression Ratio (CR) demonstrates minor differences in execution time, with a range of 50.63–54.35 s. However, the corresponding memory usage fluctuates between 0.64 and 1.14 GB, indicating a potential trade-off between time and memory. Lastly, in Scenario-D, TextRank and LexRank exhibit comparable execution times of 19.31 s and 19.01 s, respectively, while TextRank utilizes slightly less memory (0.90 GB) compared to LexRank (1.03 GB). GEATS, with an execution time of 64.02 s and memory usage of 1.08 GB, presents a distinct profile within the scenario. The outlined differences underscore the nuanced trade-offs and efficiencies inherent in each method across diverse scenarios.

4.9 GEATS-Generated Summary: A Complete Example

Table 9 illustrates the summary produced with GEATS and the original document. The GEATS-generated summary aimed to collect the essential data from the original document and present a concise representation of key details. It sought to faithfully reflect the government human rights report, addressing significant issues like revocation of citizenship, torture, travel restrictions, and social discrimination against women. The summary endeavored to capture the core content effectively, demonstrating the proficiency of the model in summarizing Arabic texts.

5 Conclusion and Future Remarks

Due to the ever-increasing volume of digitally available resources, understanding the intrinsic meaning of the tar-

Table 9 Sample GEATS-generated summary and the original document

Original Document
الدوحة ـ (اف ب) ـ اعلن مصدر رسمي ان وزير الدولة القطري للشؤون الداخلية الشيخ عبدالله بن ناصر بن خليفة آل ثاني قرر الثلاثاء اعادة تشكيل اللجنة الدائمة لشؤون الجنسية في اطار تطوير الخدمات الامنية في البلاد. وذكرت وكالة الانباء القطرية ان الشيخ عبد الله اصدر قرارا وزاريا تم بموجبه اعادة تشكيل اللجنة الدائمة لشؤون الجنسية في اطار الجهود المبذولة لتطوير الخدمات الامنية وترقية الاداء في ظل التطور الذي تشهده الوزارة والادارات الامنية. وكان تقرير لهيئة حكومية قطرية لحقوق الانسان نشر الثلاثاء، اشار الى تجاوزات في ملف حقوق الانسان بينها حالات تتعلق بسحب الجنسية وتعذيب ومنع من السفر اضافة الى استمرار التمييز الاجتماعي والاقتصادي والاسري ضد المرأة. وقال التقرير السنوي الاول من نوعه الذي اصدرته اللجنة الوطنية لحقوق الانسان الحكومية في قطر، انه تم رصد ١٤٩ شكوى والتماسا تلقتها اللجنة خلال عام ٢٠٠٤ في هذا المجال. وتحدثت مصادر عدة مؤخرا عن سحب الجنسية من عدد من القطريين في اجراء قالت السلطات القطرية انه مجرد تطبيق للقانون الذي يحظر ازدواج الجنسية في حين رد بعض من نزعت جنسيتهم الامر الى ولائهم لامير قطر السابق.
Generated Summary
وكان تقرير لهيئة حكومية قطرية لحقوق الانسان نشر الثلاثاء، اشار الى تجاوزات في ملف حقوق الانسان بينها حالات تتعلق بسحب الجنسية وتعذيب ومنع من السفر اضافة الى استمرار التمييز الاجتماعي والاقتصادي والاسري ضد المرأة. وتحدثت مصادر عدة مؤخرا عن سحب الجنسية من عدد من القطريين في اجراء قالت السلطات القطرية انه مجرد تطبيق للقانون الذي يحظر ازدواج الجنسية في حين رد بعض من نزعت جنسيتهم الامر الى ولائهم لامير قطر السابق.

get documents became a costly task. The employment of NLP-based methods and software promotes a better understanding of national attitudes toward the current economic, political, and social conflicts, and facilitates resolving customer complaints and collection of useful data. Automated text summarization is a key issue in the accomplishment of these goals. Although several studies have been conducted on automatic text summarization in common languages such as English, the studies on Arabic language are still quite limited. The present study addressed automated summarization of Arabic texts based on the extractive summarization approach principles.

There are alternative extractive text summarization methods. Graphical methods have been popular among researchers due to their ability to transfer complex structures into graphs. The present study proposed a graphical method, GEATS, due to its obvious advantages and to fill the gap in Arabic language summarization literature.

The proposed GEATS method included three major stages: pre-processing, feature extraction, and graph plotting. In the pre-processing stage, certain normalization procedures were conducted on EASC dataset documents. Certain procedures were conducted at the document level such as the removal of repeating phrases, duplicate spaces, and diacritics, while others were conducted at word level such as unification of ALEF styles and stemming. In the feature extraction phase, the popular Word2Vec method was employed to generate features about each stemmed word and to determine semantic associations between the words. The similarity between the sentences was calculated based on cosine similarity. The sentences were represented with graph nodes in the next step. After the construction of a graph that reflected all associations, the correlations between any two graph nodes that were represented by edges, were determined based on the weight of the cosine-similarity between these nodes. The final sentence score was calculated based on the PageRank algorithm score, and the sentences with high scores were included in the summary. To determine the optimal configuration and to analyze the effectiveness of the approach, certain experimental comparisons were conducted, including the analysis of GEATS performance with various feature extractors and CR parameter values, and the performance of GEATS versus other state-of-the-art methods. The findings demonstrated that the GEATS method performed better with 40% compression, when the Farasa stemmer was employed to find the word roots, word embedding was employed to extract the features, and the PageRank algorithm was applied to calculate and rank the similarities. The above-mentioned configuration of the proposed GEATS method was compared to the state-of-the-art methods based on the quality of the produced summary based on F-measures. The findings demonstrated that the GEATS method performed 7.5% better when compared to the closest alternative.

The quality of the summaries produced with the state-of-the-art methods and the proposed approach demonstrated that the GEATS method, despite its limited scope in the present study, led to a significant improvement in the automated summarization of Arabic texts; however, the field still requires further studies that would address certain issues such as consideration of additional linguistic properties to tackle the complex morphological relationships in Arabic language, employment of a particular hybrid lemmatization that utilizes both graphical and rhetorical methods, or the definition of specific stop words in documents of distinct categories found in the same dataset due to their common topic.

Acknowledgements This paper is produced from the master thesis entitled “Arabic Text Summarization Using Pagerank and Word Embedding Algorithms”.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

References

1. Versteegh, K.: The Arabic Language. Edinburgh University Press, Edinburgh (2014). https://doi.org/10.1515/9780748645299
2. Yadav, A.K.; Maurya, A.K.; Yadav, R.S.; et al.: Extractive text summarization using recent approaches: a survey. Ingénierie des Systèmes d’Information (2021). https://doi.org/10.18280/isi.260112
3. Qaroush, A.; Abu Farha, I.; Ghanem, W.; Washaha, M.; Maali, E.: An efficient single document Arabic text summarization using a combination of statistical and semantic features. J. King Saud Univ. Comput. Inf. Sci. **33**(6), 677–692 (2021). https://doi.org/10.1016/j.jksuci.2019.03.010
4. Al-Taani, A.T.; Al-Omour, M.M.: An extractive graph-based Arabic text summarization approach. In: The International Arab Conference on Information Technology, pp. 158–163 (2014)
5. Ellouze, S.; Jaoua, M.; Hadrich Belguith, L.: Arabic text summary evaluation method. In: Proceedings of the International Business Information Management Association Conference-Education Excellence and Innovation Management Through Vision 2020: From Regional Development Sustainability to Global Economic Growth, pp. 3532–3541 (2017)
6. Al-Abdallah, R.Z.; Al-Taani, A.T.: Arabic single-document text summarization using particle swarm optimization algorithm. Procedia Comput. Sci. **117**, 30–37 (2017). https://doi.org/10.1016/j.procs.2017.10.091
7. Al-Radaideh, Q.A.; Bataineh, D.Q.: A hybrid approach for Arabic text summarization using domain knowledge and genetic algorithms. Cogn. Comput. **10**, 651–669 (2018). https://doi.org/10.1007/s12559-018-9547-z
8. Al-Abdallah, R.Z.; Al-Taani, A.T.: Arabic text summarization using firefly algorithm. In: 2019 Amity International Conference on Artificial Intelligence (AICAI), pp. 61–65 (2019). https://doi.org/10.1109/AICAI.2019.8701245
9. Jaradat, Y.A.; Al-Taani, A.T.: Hybrid-based Arabic single-document text summarization approach using genatic algorithm. In: 2016 7th International Conference on Information and Communication Systems (ICICS), pp. 85–91 (2016). https://doi.org/10.1109/IACS.2016.7476091
10. Fadel, A.; Esmer, G.B.: A hybrid long Arabic text summarization system based on integrated approach between abstractive and extractive. In: Proceedings of the 2020 6th International Conference on Computer and Technology Applications. ICCTA ’20, pp. 109–114. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3397125.3397129
11. Lamsiyah, S.; El Mahdaouy, A.; El Alaoui, S.O.; Espinasse, B.: A supervised method for extractive single document summarization based on sentence embeddings and neural networks. In: Advanced Intelligent Systems for Sustainable Development (AI2SD’2019) Volume 4-Advanced Intelligent Systems for Applied Computing Sciences, pp. 75–88 (2020). https://doi.org/10.1007/978-3-030-36674-2_8
12. Al-Maleh, M.; Desouki, S.: Arabic text summarization using deep learning approach. J. Big Data **7**, 1–17 (2020). https://doi.org/10.1186/s40537-020-00386-7
13. Elbarougy, R.; Behery, G.; Khatib, A.E.: Graph-based extractive Arabic text summarization using multiple morphological analyzers. J. Inf. Sci. Eng. **36**(2), 347 (2020)
14. Etaiwi, W.; Awajan, A.: Graph-based Arabic NLP techniques: a survey. Procedia Comput. Sci. **142**, 328–333 (2018). https://doi.org/10.1016/j.procs.2018.10.488
15. Imam, I.; Nounou, N.; Hamouda, A.; Khalek, H.A.A.: An ontology-based summarization system for Arabic documents (OSSAD). Int. J. Comput. Appl. **74**(17), 38–43 (2013)
16. Waheeb, S.A.: Multi-document text summarization using text clustering for Arabic language. Ph.D. thesis, Universiti Utara Malaysia (2014). https://etd.uum.edu.my/id/eprint/4373
17. Al-Khawaldeh, F.; Samawi, V.: Lexical cohesion and entailment based segmentation for Arabic text summarization (LCEAs). World Comput. Sci. Inf. Technol. J. **5**(3), 51–60 (2015)
18. Abdulateef, S.; Khan, N.A.; Chen, B.; Shang, X.: Multidocument Arabic text summarization based on clustering and word2vec to reduce redundancy. Information (2020). https://doi.org/10.3390/info11020059
19. Elbarougy, R.; Behery, G.; El Khatib, A.: Extractive Arabic text summarization using modified pagerank algorithm. Egypt. Inform. J. **21**(2), 73–81 (2020). https://doi.org/10.1016/j.eij.2019.11.001

20. Alami, N.; Meknassi, M.; En-nahnahi, N.; El Adlouni, Y.; Ammor, O.: Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling. Expert Syst. Appl. **172**, 114652 (2021). https://doi.org/10.1016/j.eswa.2021.114652
21. Tanfouri, I.; Tlik, G.; Jarray, F.: An automatic Arabic text summarization system based on genetic algorithms. Procedia Comput. Sci. **189**, 195–202 (2021). https://doi.org/10.1016/j.procs.2021.05.083
22. Ayed, A.B.; Biskri, I.; Meunier, J.-G.: Arabic text summarization via knapsack balancing of effective retention. Procedia Comput. Sci. **189**, 312–319 (2021). https://doi.org/10.1016/j.procs.2021.05.100
23. Alshanqiti, A.; Namoun, A.; Alsughayyir, A.; Mashraqi, A.M.; Gilal, A.R.; Albouq, S.S.: Leveraging DistilBERT for summarizing Arabic text: An extractive dual-stage approach. IEEE Access **9**, 135594–135607 (2021). https://doi.org/10.1109/ACCESS.2021.3113256
24. Bahakam, O.S.; Binwahlan, M.S.F.; Mogaibel, H.A.: Statistical features and pagerank scoring fusion for arabic text summarization. In: 2022 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE), pp. 1–8 (2022). https://doi.org/10.1109/ITSS-IoE56359.2022.9990965
25. Wazery, Y.M.; Saleh, M.E.; Alharbi, A.; Ali, A.A.; et al.: Abstractive Arabic text summarization based on deep learning. Comput. Intell. Neurosci. (2022). https://doi.org/10.1155/2022/1566890
26. Reda, A.; Salah, N.; Adel, J.; Ehab, M.; Ahmed, I.; Magdy, M.; Khoriba, G.; Mohamed, E.H.: A hybrid Arabic text summarization approach based on transformers. In: 2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), pp. 56–62 (2022). https://doi.org/10.1109/MIUCC55081.2022.9781694
27. AL-Khassawneh, Y.A.; Hanandeh, E.S.: Extractive Arabic text summarization-graph-based approach. Electronics (2023). https://doi.org/10.3390/electronics12020437
28. Ko, Y.; Seo, J.: An effective sentence-extraction technique using contextual information and statistical approaches for text summarization. Pattern Recognit. Lett. **29**(9), 1366–1371 (2008). https://doi.org/10.1016/j.patrec.2008.02.008
29. Bialy, A.A.; Gaheen, M.A.; ElEraky, R.M.; ElGamal, A.F.; Ewees, A.A.: In: Abd Elaziz, M., Al-qaness, M.A.A., Ewees, A.A., Dahou, A. (eds.) Single Arabic Document Summarization Using Natural Language Processing Technique, pp. 17–37. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-34614-0_2
30. Elayeb, B.; Chouigui, A.; Bounhas, M.; Khiroun, O.B.: Automatic Arabic text summarization using analogical proportions. Cogn. Comput. **12**, 1043–1069 (2020). https://doi.org/10.1007/s12559-020-09748-y
31. Alqaisi, R.; Ghanem, W.; Qaroush, A.: Extractive multi-document Arabic text summarization using evolutionary multi-objective optimization with k-medoid clustering. IEEE Access **8**, 228206–228224 (2020). https://doi.org/10.1109/ACCESS.2020.3046494
32. Suleiman, D.; Awajan, A.: Deep learning based abstractive text summarization: approaches, datasets, evaluation measures, and challenges. Math. Probl. Eng. **2020**, 1–29 (2020). https://doi.org/10.1155/2020/9365340
33. Elbarougy, R.; Behery, G.; El Khatibm, A.: The impact of stop words processing for improving extractive graph-based Arabic text summarization. Int. J. Sci. Technol. Res. **8**(11), 2134–2139 (2019)
34. Gambhir, M.; Gupta, V.: Recent automatic text summarization techniques: a survey. Artif. Intell. Rev. **47**, 1–66 (2017). https://doi.org/10.1007/s10462-016-9475-9
35. Moratanch, N.; Chitrakala, S.: A survey on extractive text summarization. In: 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), pp. 1–6 (2017). https://doi.org/10.1109/ICCCSP.2017.7944061
36. Meena, Y.K.; Gopalani, D.: Evolutionary algorithms for extractive automatic text summarization. Procedia Comput. Sci. **48**, 244–249 (2015). https://doi.org/10.1016/j.procs.2015.04.177
37. Wills, R.S.: Google’s pagerank. Math. Intell. **28**(4), 6–11 (2006). https://doi.org/10.1007/BF02984696
38. Erkan, G.; Radev, D.R.: Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Int. Res. **22**(1), 457–479 (2004). https://doi.org/10.1613/jair.1523
39. Mihalcea, R.; Tarau, P.: Textrank: Bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004). https://aclanthology.org/W04-3252.pdf
40. Page, L.; Brin, S.; Motwani, R.; Winograd, T.: The pagerank citation ranking: Bring order to the web. Technical report, Stanford University (1998)
41. Cohen, R.; Havlin, S.; ben-Avraham, D.: 4. Structural properties of scale-free networks, pp. 85–110 (2002). https://doi.org/10.1002/3527602755.ch4
42. Brin, S.; Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**(1), 107–117 (1998). https://doi.org/10.1016/S0169-7552(98)00110-X
43. Attia, M.A.: Handling Arabic morphological and syntactic ambiguity within the lfg framework with a view to machine translation. Ph.D. thesis, The University of Manchester (United Kingdom) (2008)
44. El-Khair, I.A.: Effects of stop words elimination for Arabic information retrieval: a comparative study. arXiv preprint arXiv:1702.01925 (2017)
45. Abdelali, A.; Darwish, K.; Durrani, N.; Mubarak, H.: Farasa: a fast and furious segmenter for Arabic. In: DeNero, J., Finlayson, M., Reddy, S. (eds.) Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pp. 11–16. Association for Computational Linguistics, San Diego, California (2016). https://doi.org/10.18653/v1/N16-3003
46. El-Haj, M.; Kruschwitz, U.; Fox, C.: Using mechanical turk to create a corpus of Arabic summaries (2010)
47. Powers, D.M.: Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061 (2020)
48. Lin, C.-Y.: Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004). https://aclanthology.org/W04-1013.pdf
49. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.-J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL ’02, pp. 311–318. Association for Computational Linguistics, USA (2002). https://doi.org/10.3115/1073083.1073135
50. Achananuparp, P.; Hu, X.; Shen, X.: The evaluation of sentence similarity measures. In: Data Warehousing and Knowledge Discovery: 10th International Conference, DaWaK 2008 Turin, Italy, September 2–5, 2008 Proceedings 10, pp. 305–316. Springer (2008). https://doi.org/10.1007/978-3-540-85836-2_29