Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi
Pamukkale University Journal of Engineering Sciences

An unsupervised hybrid model for keyphrase extraction

Anahtar kelime çıkarımı için denetimsiz hibrit bir model

Özlem Örnek[1], Efnan Şora Günal[1], Eyyüp Gülbandılar[1]*

[1]Department of Computer Engineering, Eskişehir Osmangazi University, Eskişehir, Türkiye
ozlemmornek@gmail.com, esora@ogu.edu.tr, egunbandilar@ogu.edu.tr

Received/Geliş Tarihi: 22.01.2025 Revision/Düzeltme Tarihi: 11.06.2025 doi: 10.5505/pajes.2025.05400
Accepted/Kabul Tarihi: 22.07.2025 Research Article/Araştırma Makalesi

Abstract

Extracting the pertinent words from a text can be defined as keyphrase or keyword extraction. While a keyphrase consists of multiple words and a keyword is a single word, they can also be used interchangeably. Though there are different methods for keyword extraction in the literature, unsupervised methods come to the fore with their independence from the domain and not needing training with labeled data. Hence, in this work, a new unsupervised hybrid model is presented for the keyphrase extraction task. The proposed model consists of a graph-based and an embedding-based method. The proposed model is developed using the graph centrality criteria and the skip-gram embedding method created for each document. The model was evaluated on a dataset and compared with the literature. Following comprehensive experiments, it was observed that our model provided comparable performance with statistical models, while outperforming other graph-based and embedding-based models.

Keywords: Keyphrase extraction, Unsupervised learning, Hybrid model, Graph-based method, Word embedding.

Öz

Metinden ilgili sözcükleri çıkarmak, anahtar sözcük öbeği veya anahtar sözcük çıkarma olarak tanımlanabilir. Bir anahtar sözcük öbeği birden fazla sözcükten oluşurken, bir anahtar sözcük tek bir sözcük olsa da bunlar birbirinin yerine kullanılabilir. Literatürde anahtar sözcük çıkarma için farklı yöntemler bulunmasına rağmen, denetimsiz yöntemler, alandan bağımsız olmaları ve etiketli verilerle eğitim gerektirmemeleri nedeniyle ön plana çıkmaktadır. Bu nedenle, bu çalışmada, anahtar sözcük öbeği çıkarma görevi için yeni bir denetimsiz hibrit model sunulmuştur. Önerilen model, çizge tabanlı ve gömme tabanlı yöntemlerden oluşmaktadır. Önerilen model, çizge merkezilik ölçütleri ve her bir belge için oluşturulan skip-gram gömme yöntemi kullanılarak geliştirilmiştir. Model, bir veri kümesi üzerinde değerlendirilmiş ve literatürle karşılaştırılmıştır. Kapsamlı deneyler sonucunda, modelimizin istatistiksel modellerle karşılaştırılabilir performans sağladığı, diğer çizge tabanlı ve gömme tabanlı modellerden daha iyi performans gösterdiği gözlenmiştir.

Anahtar kelimeler: Anahtar kelime çıkarma, Denetimsiz öğrenme, Hibrit model, Graf tabanlı yöntem, Kelime gömme.

1 Introduction

Keywords are important in semantic indexing, document clustering, and classification in digital information management systems, because indicate the concepts about topic. Keywords provide information about the fundamental topics of the document for readers and helping them decide to continue reading or not the document [1]. According to Papagiannopoulou and Tsoumakas [2], keywords can be used for document indexing, clustering, and classification, guiding automatic document summarization systems, recommending new articles or books to users in the academic publishing industry, highlighting missing citations to authors and analysis of content trends. Merrouni, Frikh, and Ouhbi [1] have given document clustering, document summarization, information retrieval (IR) systems, document indexing, web mining, opinion mining, search engines, recommendation systems, ontology, information extraction, and topic analysis as examples of some uses of keywords in text mining, natural language processing (NLP) and IR.

The keywords obtainment can be examined in two categories: assignment and extraction. In assignment category, keywords are selected from a controlled dictionary of terms or a learned model, and documents are classified depending on their content [3]. On the other hand, keyword extraction is the textual information processing task of automatically extracting phrases that can express all the essential aspects of a document's content [2]. On the other hand, the purpose of keyword extraction is to specify a word or phrase that represents the essential content of the text [4], [5]. There are some difficulties in extracting keywords. In the study of Hasan and Ng [6], the factors affecting the difficulty of keyword extraction studies were stated as the length of the document, the structural consistency that may vary according to the document type, the change in the subject, and the correlation of the subject. In the literature, there are methods for keyword extraction such as supervised and unsupervised. Though the supervised methods offer relatively higher performance, they are domain-dependent and need to labeled training data. Therefore, the unsupervised methods gain more attraction as they are independent from the domain and not needing training with labeled data.

Leveraging the advantages of unsupervised learning while addressing the scarcity of hybrid approaches, this work proposes a new model for automatic keyphrase extraction. Our model integrates graph-based methods with basic natural language processing, deliberately using a core set of essential metrics selected for their foundational characteristics. The primary contribution is demonstrating that this streamlined approach can achieve performance comparable to more complex models, establishing a robust and fundamental baseline.

The proposed model consists of two different methods: graph-based and embedding-based. The model is developed using the

*Corresponding author/Yazışılan Yazar

graph centrality criteria, including degree, inner degree, outer degree, closeness, and betweenness associated with the number of words or phrases in the text, and the skip-gram embedding method created for each document. The model was evaluated on a widely used benchmark dataset and compared with the literature. Experimental results revealed that our model provided comparable performance with statistical models, while outperforming other graph-based and embedding-based models.

The remainder of the paper is organized as follows: Section 2 reviews the recent literature on keyphrase extraction. Section 3 presents the proposed method, followed by the experimental work and results in Section 4. Finally, Section 5 discusses the findings and concludes the paper.

2 Literature review

As mentioned earlier, for the keyword extraction task, unsupervised methods get more attention as independent from the domain and not needing training with labeled data, unlike supervised methods. An unsupervised keyword extraction models commonly consists of several stages, including preprocessing, the identification of candidate keywords, and the selection of the candidate keywords based on a scoring mechanism [2], [6].

In the literature, unsupervised methods available for keyword extraction can be divided as: statistical, embedding-based, language model-based, and graph-based.

For the statistical methods, TFIDF is the common baseline. KP-Miner, KeyCluster, YAKE algorithms are just some examples of those methods. KP-Miner [7] is a keyword extraction system that uses TF and IDF scores and various statistical information (e.g., term position) [2]. KeyCluster [8] tries to find keywords for document to show the main topics [2]. In the KeyCluster, stop words are removed and candidate terms are selected. Then, its groups by spectral clustering. In clustering, its used Wikipedia-based or co-occurrence-based metrics for calculate the semantic relationship of candidate terms. Finally, it finds sample terms from each set to extract keywords from the document. YAKE uses the location/frequency of the term, as well as context information and statistical measures such as the spread of terms throughout the document.

Embedding methods came to the fore with [9] that presents the continuous bag of words model (CBOW) and the continuous skip-gram model. Also, sentence embeddings such as Doc2Vec [10], Sent2Vec [11] as well as the Global Vectors (GloVe) [12] are employed to extract keywords.

In language model-based methods, commonly, an n-gram language model used for assigning a probability value to different sequence of words. For example, Tomokiyo and Hurst [13] used unigram and n-gram language models on a different corpus.

In the graph-based method, the basic idea is to create a graph containing candidate expressions in a document as nodes, where each edge connects the relevant candidate keywords. The purpose can be explained as scoring and ranking the nodes of the resulting graph and creating keywords [2]. Graph-based methods can be given in four groups [2]: graph statistics, information from neighbors or citation networks, topic-based, and semantic information. There are studies in this field in the literature [14]-[16]. Some examples to the graph-based methods are as follows:

Kumar, Srinathan, and Varma [17] performed keyword extraction using n-gram filtering technique, statistical feature (weighted centrality scores of words), and co-location power (nearest neighbor calculated using Dbpedia texts). Ying et al. [18] created graphs from word to word, sentence to sentence, and word to sentence, and then used graph sorting algorithm and term clustering (K-means). The method has been tested on Hulth2003 and 500N datasets. Song et al. [19] proposed a contextual keyword extraction method for meeting transcripts using TextRank. Li et al. [20] used dictionary and HMM based method to divide scientific summaries into words. They proposed a multi-word keyword generation method using TextRank with location restrictions and statistical features. The method was tested using Chinese scientific summaries. Florescu and Caragea [21] proposed a new sentence scoring scheme for unsupervised keyword extraction. The method has been tested with Nguyen, WWW, and KDD datasets. Batsuren, et al. [22] developed a method using dependency graph, anti-patterns (the candidates, which are not keywords), TextRank and Stanford dependency parser (SDP), and the developed method was tested on the Inspec [23] dataset. Biswas et al. [24] suggested the Keyword Extraction using Collective Node Weight (KECNW) method. In this method, the features such as selectivity centrality, importance of neighboring nodes, distance to the central node, location of a node and term frequency are used for node weight assignment, while node edge order calculation and degree centrality calculation are used for keyword extraction. The method has been tested using Twitter data. Mothe et al. [25] tested the effect of graph-based methods and word insertion using the INSPEC, SEMEVAL, and BIOMED datasets for different methods. Vega-Oliveros et al. [26] analyzed polycentricity index, nine centrality criteria (clustering coefficient, closeness, betweenness, degree, eccentricity, page rank, structural holes, eigenvector, k-core) and used the clustering algorithms (DBSCAN, Expectation-maximization (EM) and K- means). The method has been tested on Hulth2003, Marujo2012, and SemEval2010 datasets. Thushara et al. [27] tested TF-IDF, graph-based model (KECNW) and sentence embedding (EmbedRank) methods with Twitter data and Hulth2003, 500N, Inspec, Nguyen, and DUC 2001 datasets. Li et al. [28] tested graph-based sorting and subject-based clustering methods with WWW, KDD, GSA, and ACM datasets. Zhang et al. [29] conducted an experimental study on TextRank using Hulth2003 and Krapivin2009 datasets to test the effectiveness of different parameter settings of TextRank.

Brin and Page's [30] PageRank study which is based on eigenvector centrality used in graph-based keyword extraction. TextRank [31] is the first graph-based keyword extraction method [2]. In TextRank, text is first tokenized, and POS tags are obtained. Then, graph is building using nouns and adjectives. An edge is added between the nodes (nouns or adjectives) found together in the specified size word window. The resulting graph is undirected and weightless. Then, the score is calculated using the PageRank algorithm. SingleRank is an extension of TextRank that includes weights on the edges [2]. Edge weight is specifying with the count of two words occur together. The words scores are added up for each noun and adjective in the text document, and the top scored are returned as keywords.

ExpandRank [32], based on SingleRank, considers neighboring documents in the same dataset when extracting keywords from a document. The Rapid Automatic Keyword Extraction (RAKE) method uses rank and frequency to give scores to word phrases

[2], [33]. RAKE creates a graph of word-word associations, and for each candidate phrase, a score is assigned with the calculation of addition of the scores for the words. Finally, the top candidate phrases according to their score values are selected as the keywords for the given document. SGRank [34] and PositionRank (PR) [35] use positional, statistical and word co-occurrence information [2]. SGRank firstly obtains possible n-grams from the input text, removing punctuation from words, and eliminating the words that are different from nouns, adjectives, or verbs. Secondly, candidate n-grams are sorted according to an altered version of TF-IDF. Then, the best scored candidates are re-ranked according to additional statistical methods, such as the first appearance position. Finally, the produced ranking is included in an algorithm that generates the keyword candidates final ranking. PositionRank (PR) is an unsupervised graph-based method to catch repeated expressions in view of word-to-word links and the corresponding positions of words in the text. It includes positions of a word in a bias-weighted PageRank. Finally, keywords are scored and ranked. After the completion of all stages of the keyword extraction, the keywords obtained can be examined and the errors of the keyword extractor can be detected [2]. In the study of Hasan and Ng [6], it was stated that the errors originating from the keyword extraction methods are evaluation errors, redundancy errors, sparsity errors, and overproduction errors.

In the literature also some unsupervised hybrid methods exist. These methods mostly consist of combinations of the statistical, embedding-based, language model-based, or graph-based methods [36], [37]. Sarracén et al. [36], used language model-based and graph-based methods.

3 Materials and methods

In this section, graph-based keyword extraction is first briefly explained. Then, the proposed hybrid model is introduced.

3.1 Graph-based keyword extraction

The steps of keywords extraction from a document using the graphs are shown in Figure 1 [38],[39].

Co-occurrence relationships (linking neighboring words that occur together in a settled size window in the text or combining all words found jointly in a sentence, paragraph, chapter, or document), syntax relationships (linking words by their relationship in a syntax dependency graph), and semantic relations (bindings the words with alike intent, words written alike but having different meanings, synonyms, antonyms, synonyms, etc.) can be used to determine the edges between nodes in graphs [38], [39].

The centrality defines the importance or properties of the nodes in a graph [38]. These are also used in approaches for extracting the keyword. Common centrality measures are degree (the count of the neighbors for a node), degree centrality, in-degree centrality, out-degree centrality, out-strength, in-strength, in-selectivity, out-selectivity, closeness centrality, betweenness centrality, and eigenvector centrality [39].

In our work, the degree of a node, in-degree centrality, out-degree centrality, closeness centrality, and betweenness centrality are used. The centrality degree of a node is calculated using the Equation in 1, where d_v is the degree of the node v (the sum of the number of nodes coming to and leaving from the node v), and N is the number of all nodes found in the graph.

$$Node\ Centrality\ Degree(v) = d_v/(|N| - 1) \quad (1)$$

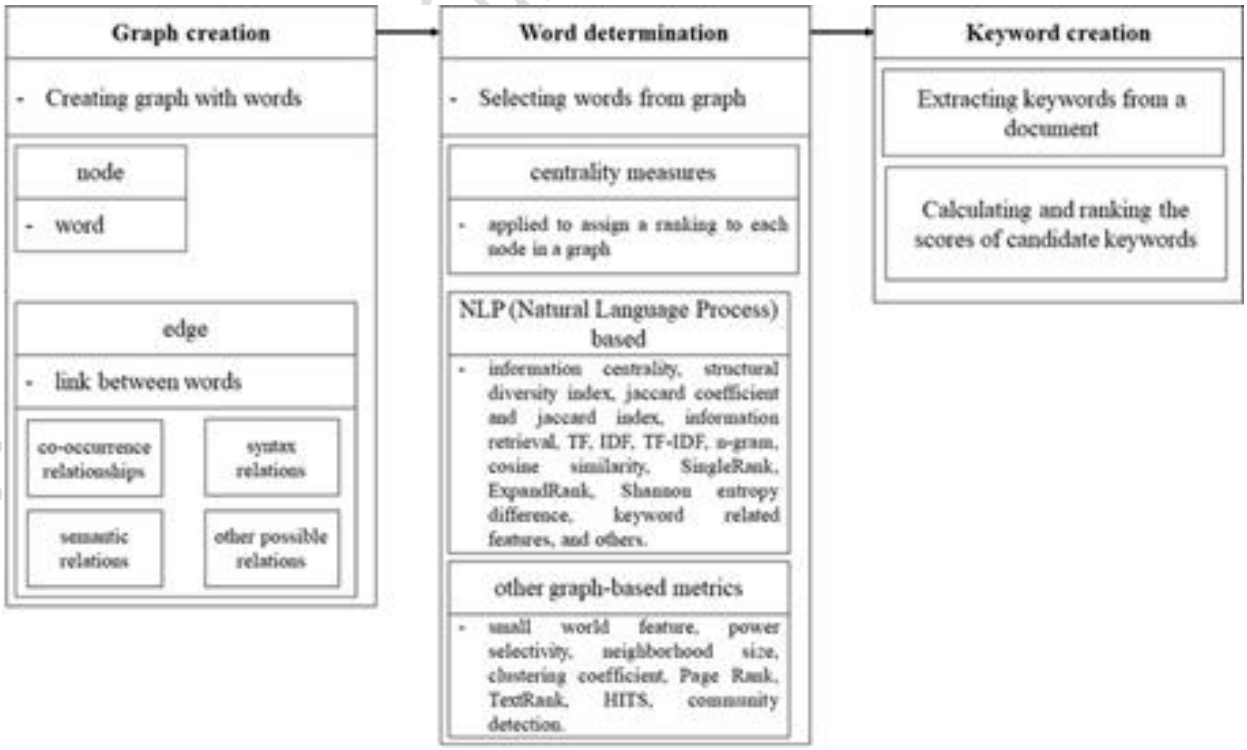

Figure 1. The stages of extracting keywords from a document with the graph-based approach.

The out-degree centrality is calculated using the Equation in 2, where d_v^{out} is the out-degree of the node v (the number of nodes leaving the node v).

$$Node\ Out-Degree(v) = d_v^{out}/(|N| - 1) \quad (2)$$

The in-degree centrality is calculated using the Equation 3, where d_v^{in} is the in-degree of the node v (the number of nodes arriving at the node v).

$$Node\ In-Degree(v) = d_v^{in}/(|N| - 1) \quad (3)$$

The closeness centrality is calculated using the Equation in 4, where R(v) is the set of all nodes that the node v has access to, and d(v, u) is the path length between u (a node in the cluster R(v)) and the node v.

$$Node\ Closeness\ Centrality(v) = \left(\frac{|R(v)|}{|N|-1}\right) \times \left(\frac{|R(v)|}{\sum_{u \in R(v)} d(v,u)}\right) \quad (4)$$

The betweenness centrality is calculated using the Equation in 5 and 6, where $\sigma_{s,t}$ is the number of the shortest paths between the nodes s and t, $\sigma_{s,t}(v)$ is the number of the shortest paths between the nodes s and t through the node v.

$$Node\ Betweenness\ Centrality(v) = \sum_{s,t \in N} (\sigma_{s,t}(v)\ /\sigma_{s,t}) \quad (5)$$

$$Normalized\ Node\ Betweenness\ Centrality\ for\ Directed\ Graph(v) = \frac{\sum_{s,t \in N}\left(\frac{\sigma_{s,t}(v)}{\sigma_{s,t}}\right)}{(|N| - 1) \times (|N| - 2)} \quad (6)$$

3.2 Proposed model

The proposed model utilizes the basic graph methods which are described previously and the "Word2Vec" skip-gram. To learn word associations from a corpus of text, Word2vec uses a neural network model. It can detect synonyms or suggest words with trained model. Word2vec represents each different word by vector. The semantic similarity grade between these vectors represented with cosine similarity. The skip-gram uses the window around a word to predict its context words. And skip-gram places more importance on nearby words than on farther words [9], [40]. The proposed model block diagram is given in Figure 2.

In the first stage of the proposed model, a document is first selected from the document collection. Then, sentence segmentation is performed on the selected document. The resulting sentences are forwarded to the second stage.

In the second stage, several pre-processing methods are applied to the sentences. The methods include lowercase conversion, removal of numerals, removal of special characters, removal of stop words, removal of extra spaces, and lemmatization.

In the third stage, the skip-gram model is created. For this purpose, all sentences are combined separately.

In the fourth stage, a graph is created. Firstly, n-gram is used to get the occurrence relationships between words. Then, the acquired relationship is used to provide the connections (edges) between words. In the graph, the nodes correspond to words, whereas edges indicate the connection between words. For n-gram occurrence relationships, n-grams (n =3) are used for each sentence. Graph nodes and edge connections between them are realized using n-gram relations for each sentence.

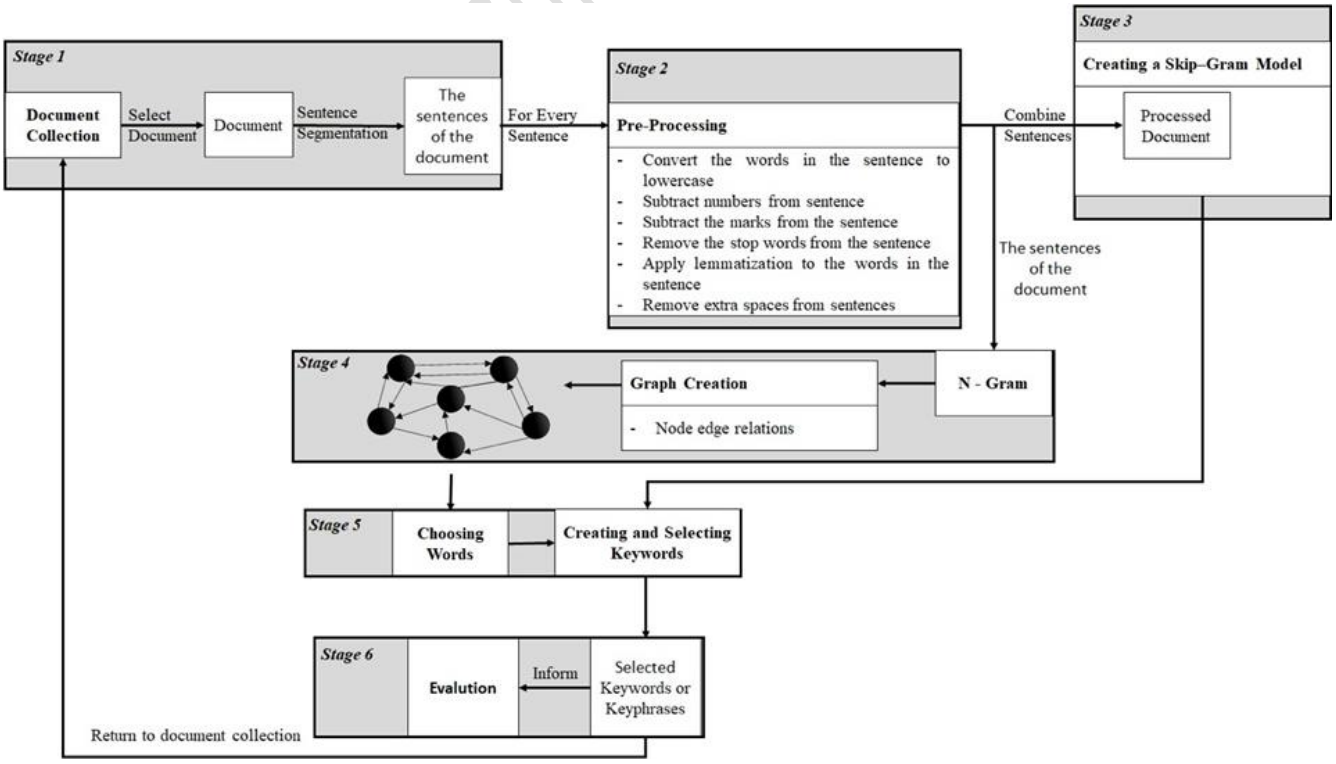

Figure 2. The proposed model block diagram.

In the fifth stage, keywords are selected. Firstly, the nodes (words) with high keyword generation potential are identified. For this purpose, the first 30 nodes with the largest out-degree in the graph are chosen. Then, for all nodes in the graph, degree, out-degree, in-degree, closeness, and betweenness centralities are calculated. The cases, where each node within the selected 30 nodes can reach all nodes in the graph with a maximum of 2 steps (i.e., reaching at most 2 edges at the other nodes) is obtained with the "NetworkX" simple path function of the Python package. Next, a score for each candidate keyword is calculated using the Equations in 7 and 8 for all the combinations obtained.

Finally, the candidate keywords are ranked in descending order of their scores, and the top K keywords are selected from the ordered list.

Specifically, seven different calculation approaches are used to calculate the scores of the candidate keyphares based on the number of words in the keyphrase. The calculations are based on degree, betweenness, and proximity values. The calculation algorithm is provided in Algorithm 1.

$$score = \left(count(N_S) \times \left(D^-(v_1) + \frac{D^+(v_1)}{D^-(v_1)} \right) \right) \times approach \quad (7)$$

$$score = \left(count(N_S) + \frac{ave_{similarity}}{i} \right) \times approach \quad (8)$$

Algorithm 1 Calculation of the scores of the candidate keyword(s)

Definitions: $D^-(v_i) = v_i$ node out centrality t
$D^+(v_i) = v_i$ node in centrality t
$D(v_i) = v_i$ node degree t
$Y(v_i) = v_i$ node closeness centrality t
$NA(v_i) = v_i$ node normalized betweenness centrality t
N_S = Nodes for word(s) ($v_i \in N_S$)
i = Number of nodes in N_S
$score = 0$: The scores of the candidate keyword(s)
$calculation\_approach$: One of the approaches to get scores for candidate keyword(s) t
$approach$ = Selected calculations approach value
$count(N_S)$ = Number of times node(s) occur in the text together
$similarity(v_j, v_k)$ = Similarity value between v_j and v_k nodes according to the skip-gram model (if one of the nodes is not found in the model, the value will be 0).
if $i = 1$ **then**
  if $calculation\_approach = degree$ **then** $approach = D(v_1)$
  else if $calculation\_approach = closeness$ **then** $approach = Y(v_1)$
  else if $calculation\_approach = betweenness$ **then** $approach = NA(v_1)$
  else if $calculation\_approach = degree + closeness$ **then**
    $approach = ((D(v_1) + Y(v_1))$
  else if $calculation\_approach = degree + closeness + betweenness$ **then**
    $approach = (D(v_1) + Y(v_1) + NA(v_1))$
  else if $calculation\_approach = degree + closeness - betweenness$ **then**
    $approach = ((D(v_1) + Y(v_1)) - NA(v_1))$
  else if $calculation\_approach = closeness - betweenness$ **then**
    $approach = (Y(v_1) - NA(v_1))$
  end if
  $score = (count(N_S) \times (D^-(v_1) + D^+(v_1) / D^-(v_1))) \times approach.$
else if $i > 1$ **then** $ave\_similarity = 0$
  if $calculation\_approach = degree$ **then**
    $approach = \sum_{v_i \in N_S} D(v_i)$
  else if $calculation\_approach = closeness$ **then**
    $approach = \sum_{v_i \in N_S} Y(v_i)$
  else if $calculation\_approach = betweenness$ **then**
    $approach = \sum_{v_i \in N_S} NA(v_i)$
  else if $calculation\_approach = degree + closeness$ **then**
    $approach = \sum_{v_i \in N_S} (D(v_i) + Y(v_i))$
  else if $calculation\_approach = degree + closeness + betweenness$ **then**
    $approach = \sum_{v_i \in N_S} (D(v_i) + Y(v_i) + NA(v_i))$
  else if $calculation\_approach = degree + closeness - betweenness$ **then**
    $approach = \sum_{v_i \in N_S} (D(v_i) + Y(v_i) - NA(v_i))$
  else if $calculation\_approach = closeness - betweenness$ **then**
    $approach = \sum_{v_i \in N_S} (Y(v_i) - NA(v_i))$
  end if
  for $j = 1, 2, \ldots i$ **do**
    for $k = j + 1, \ldots i$ **do**
      $ave\_similarity = ave\_similarity + similarity(v_j, v_k)$
    end for
  end for
  $score = (count(N_S) + ave\_similarity / i) \times approach$
end if

Finally, in the sixth stage, the selected keywords are evaluated. For the evaluation, stop words (if present) are removed from the selected keywords, and stemming is applied. Then, performance metrics as recall, precision, and F1 score are calculated with respect to the precise match evaluation of the resulting keywords of the proposed model against the ground truth keyword list.

4 Experimental work

In this section, firstly the description of the used dataset to evaluate the proposed model. Then, the results of the experimental work are given.

4.1 Dataset

In our study, the SemEval2010 dataset [41], which is frequently preferred in the related literature, was used. The dataset contains keywords tagged by both the authors and the readers. The dataset contains 244 documents in English, consisting of technical and scientific full-text documents in the areas of distributed systems, distributed artificial intelligence, social and behavioral sciences, multi-agent systems, economics, information search and retrieval. The keyword/document ratio in the dataset is 15.

4.2 Results

Recall, precision and F1 are among the most common success metrics to evaluate performance in text mining studies. These metrics are also used in keyword extraction studies and exact match evaluation. The purpose of the precise match evaluation is to determine the correct match of the predetermined target keywords for the documents in the dataset with the keywords determined using the developed algorithm. Precision, recall, and F1-metrics are formulated as follows,

$$precision = \frac{number\ of\ correct\ matches}{number\ of\ predicted} = \frac{TP}{(TP + FP)} \quad (9)$$

$$recall = \frac{number\ of\ correct\ matches}{number\ of\ actual\ values} = \frac{TP}{(TP + FN)} \quad (10)$$

$$F_{1-measure} = 2 \times \frac{precision \times recall}{precision + recall} \quad (11)$$

TP, FP and FN respectively represent the number of true positives, the number of false positives and the number of false negatives.

Before calculating these metrics, "Stemming" is first applied to selected keywords. Secondly, stop-words are eliminated from the given keywords in the dataset, and "stemming" is applied. Then, precision, recall, and F1 metrics are calculated with respect to the precise match evaluation of the selected keywords using the proposed model and the keywords previously tagged by the authors and readers for the related dataset. In evaluating keyword extraction algorithms, calculations of the success metrics are usually carried out for the best 5, 10, 15, and 20 keywords. F1 scores of the proposed model on the SemEval2010 dataset for 10, 15, and 20 keywords are listed in Table 1. The F1 scores provided in the following tables represent the F1 metric calculated separately for 10, 15 and 20 keywords extracted from each document in the SemEval2010 dataset, respectively.

Table 1. F1 scores of the proposed model for 10, 15, and 20 keywords.

Calculation Approach	F1@10	F1@15	F1@20
Degree	0.152	0.154	0.149
Closeness	**0.170**	**0.171**	**0.169**
Betweenness	0.14	0.144	0.139
Degree + Closeness	0.164	**0.171**	0.166
Degree + Closeness + Betweenness	0.162	0.167	0.165
Degree + Closeness - Betweenness	**0.170**	0.170	0.168
Closeness - Betweenness	0.168	**0.173**	**0.169**

Using closeness in the calculation, 10 keywords extracted for a document in the SemEval2010 dataset, and the corresponding authors and readers keywords for the document in the dataset are given in Table 2. In the table, the keywords listed as given in their stem forms.

Table 2. Keywords of the proposed model vs. authors and readers.

Keywords of the proposed model	Keywords of the authors and readers
uddi registri	grid servic discoveri
servic	**uddi**
uddi	distribut web-servic discoveri architectur
proxi registri	dht base uddi registri hierarchi
servic discoveri	deploy issu
uddi key	bamboo dht code
web servic	case-insensit search
grid servic	queri
servic name	longest avail prefix
grid comput	qos-bas servic discoveri
	autonom control
	uddi registri
	scalabl issu
	soft state
	dht
	web servic
	grid comput
	md
	discoveri

In Table 2, the keywords in bold indicated that exactly match in the dataset. The representative graph image of result created based on proposed model is given in Figure 3. The representative graph image has been created with graph class in [42]. It can be seen from Table 2 that "web service" and "grid comput", which are in bold, provide an exact match. However, "grid servic discovery", one of the keywords given in the dataset, which was not selected with the proposed model, is also available in Figure 3.

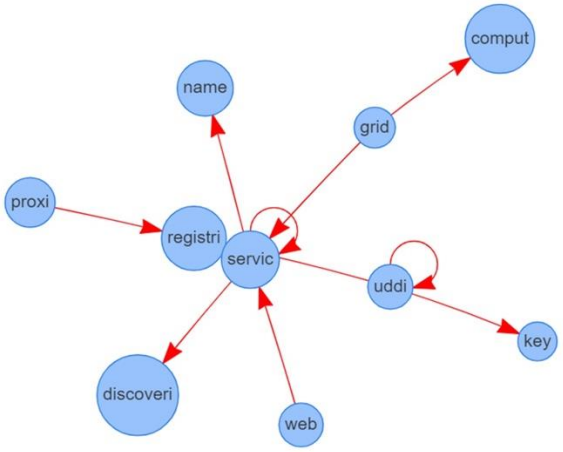

Figure 3. Graph representation of the selected keywords using the proposed model for a sample document.

5 Discussions and conclusions

In our work, an unsupervised hybrid model, which utilizes both graph-based and embedding-based approaches, is proposed to extract keywords automatically. The proposed method was evaluated on the SemEval2010 [41] benchmark dataset frequently used in the literature and an F1 score of up to 0.173 is achieved. The performance of our model was also compared with the literature based on the exact match evaluation. The summary of comparison is given in Table 3, where the category, description, and F1 scores for different numbers of keywords of each method are explicitly given. It is obvious from the table that our model outperforms other graph-based and embedding-based models. The use of different computational approaches for different combinations of graph centrality criteria (degree, closeness, betweenness) and examining the contribution of these criteria to the extraction performance can be given as the other contributions of our work. The experimental results revealed that "degree of closeness" has a positive effect while "degree of betweenness" has a negative effect on the performance. It can be also stated that "degree" contributed positively when used together with "closeness", but its use alone is insufficient.

In future work, the improvement of the proposed model or the use of feature selection methods in keyword selection can be studied.

6 Author contribution statements

In this study, the Author 1 developed the theory and performed the computations. Author 2 and Author 3 encouraged Author 1 to peruse the research topic and inspected the findings of this work. Author 1 composed the manuscript with support from Author 2 and Author 3. All authors discussed the results and contributed to the final manuscript.

7 Ethics committee approval and conflict of interest statement

There is no need to obtain permission from the ethics committee for the article prepared. There is no conflict of interest with any person / institution in the article prepared.

Table 3. Comparison of the performance of the proposed model with the literature.

Reference	Category of the Method	Description of the Method	F1@10	F1@15	F1@20
[43]	Embedding-based	Distributed skip-gram model	0.155	0.159	-
[44]	Graph-based	Use of structural and semantic information	0.167	-	-
		Rapid Automatic Keyword Extraction (RAKE)	0.114	-	0.147
[45]	Hybrid	TopicRank (TR) + Yet Another Keyphrase Extraction (YAKE) + RAKE	-	0.33	0.26
[46]	Embeddings-based	YAKE!	0.113	-	-
		KPMiner	0.202	-	-
		Word Attraction Rank	0.027	-	-
		EmbedRank	0.026	-	-
		Key2Vec	0.057	-	-
		SIFRank +	0.1	-	-
		KPRank	0.016	-	-
		KeyBERT	0.003	-	-
[47]	Unsupervised	SWaP	0.142	-	-
[2]	Graph-based	MultipartiteRank (MR)	0.146	-	0.161
		TopicRank (TR)	0.134	-	0.142
		PositionRank (PR)	0.131	-	0.127
		SingleRank (SR)	0.036	-	0.053
[2]	Embedding-based	Reference Vector Algorithm (RVA)	0.096	-	0.125
Proposed Model	Hybrid (Graph-based and Embedding-based)	Combination of graph-based and skip-gram models	0.170	0.173	0.169

8 References

[1] Merrouni ZA, Frikh B, Ouhbi B. “Automatic keyphrase extraction: a survey and trends”. *Journal of Intelligent Information Systems*, 54(2), 391–424, 2019.

[2] Papagiannopoulou E, Tsoumakas G. “A review of keyphrase extraction”. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2), 1-45, 2019.

[3] Beliga S. "Keyword extraction: a review of methods and approaches". *University of Rijeka, Department of Informatics*, 1(9), 1-9, 2014.

[4] Bougouin A, Boudin F, Daille B. "Topicrank: Graph-based topic ranking for keyphrase extraction". *International joint conference on natural language processing (IJCNLP)*, Nagoya, Japan, 14-18 October 2013.

[5] Sun C, Hu L, Li S, Li T, Li H, Chi L. “A Review of Unsupervised Keyphrase Extraction Methods Using Within-Collection Resources”. *Symmetry*, 12(11), 1-20, 2020.

[6] Hasan KS, Ng V. "Automatic keyphrase extraction: A survey of the state of the art". *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore/Maryland, USA, 23-25 June 2014.

[7] El-Beltagy SR, Rafea A. “KP-Miner: A keyphrase extraction system for English and Arabic documents”. *Information Systems*, 34(1), 132–144, 2009.

[8] Liu Z, Li P, Zheng Y, Sun M. “Clustering to Find Exemplar Terms for Keyphrase Extraction”. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, 6-7 August 2009.

[9] Mikolov T, Chen K, Corrado G, Dean J. “Efficient Estimation of Word Representations in Vector Space”. *International Conference on Learning Representations*, Scottsdale/Arizona, USA, 2-4 May 2013.

[10] Lau JH, Baldwin T. “An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation”. *Proceedings of the 1st Workshop on Representation Learning for NLP*, Berlin, Germany, 11 August 2016.

[11] Pagliardini M, Gupta P, Jaggi M. “Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features”. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans/Louisiana, USA, 1 – 6 June 2018.

[12] Pennington J, Socher R, Manning CD. "Glove: Global vectors for word representation". *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar, 25-29 October 2014.

[13] Tomokiyo T, Hurst M. "A language model approach to keyphrase extraction". *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment*, Sapporo, Japan, 12 July 2003.

[14] Chi L, Hu L. “ISKE: An unsupervised automatic keyphrase extraction approach using the iterated sentences based on graph method”. *Knowledge-Based Systems*, 223, 107014-107026, 2021.

[15] Zhao L, Miao Z, Wang C, Kong W. “An Unsupervised Keyword Extraction Method based on Text Semantic Graph”. *2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Beijing, China, 03-05 October 2022.

[16] Liao S, Yang Z, Liao Q, Zheng Z. “TopicLPRank: a keyphrase extraction method based on improved TopicRank”. *The Journal of supercomputing/Journal of supercomputing*, 79(8), 9073–9092, 2023.

[17] Kumar N, Srinathan K, Varma V. “A graph-based unsupervised N-gram filtration technique for automatic keyphrase extraction”. *International Journal of Data Mining, Modelling and Management*, 8(2), 124-143, 2016.

[18] Ying Y, Qingping T, Qinzheng X, Ping Z, Panpan L. “A Graph-based Approach of Automatic Keyphrase Extraction”. *Procedia Computer Science*, 107, 248–255, 2017.

[19] Song HJ, Go J, Park SB, Park SY, Kim KY. “A just-in-time keyword extraction from meeting transcripts using temporal and participant information”. *Journal of Intelligent Information Systems*, 48(1), 117–140, 2016.

[20] Li SQ, Du SM, Xing XZ. “A Keyword Extraction Method for Chinese Scientific Abstracts”. *Proceedings of the 2017 International Conference on Wireless Communications, Networking and Applications*, Shenzhen, China, 20-22 October 2017.

[21] Florescu C, Caragea C. “A New Scheme for Scoring Phrases in Unsupervised Keyphrase Extraction”. *Advances in Information Retrieval: 39th European Conference on IR Research*, Aberdeen, UK, 8-13 April 2017.

[22] Batsuren K, Batbaatar E, Munkhdalai T, Li M, Namsrai OE, Ryu KH. "A Dependency Graph-Based Keyphrase Extraction Method Using Anti-patterns". *Journal of Information Processing Systems*, 14(5), 1254-1271, 2018.

[23] Hulth A. “Improved automatic keyword extraction given more linguistic knowledge”. *Proceedings of the 2003 conference on Empirical methods in natural language processing*, Sapporo, Japan, 11-12 July 2003.

[24] Biswas SK, Bordoloi M, Shreya J. “A graph based keyword extraction model using collective node weight”. *Expert Systems with Applications*, 97, 51–59, 2018.

[25] Mothe J, Ramiandrisoa F, Rasolomanana M. "Automatic keyphrase extraction using graph-based methods". *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, Pau, France, 9 – 13 April 2018.

[26] Vega-Oliveros DA, Gomes PS, Milios EE, Berton L. “A multi-centrality index for graph-based keyword extraction”. *Information Processing & Management*, 56(6), p. 102063-102080, 2019.

[27] Thushara MG, Anjali S, Nai MM. "An Analysis on Different Document Keyword Extraction Methods". *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 27-29 March 2019.

[28] Li TF, Hu L, Chu JF, Li HT, Chi L. “An Unsupervised Approach for Keyphrase Extraction Using Within-Collection Resources”. *IEEE Access*, 7, 126088–126097, 2019.

[29] Zhang M, Li X, Yue S, Yang L. “An Empirical Study of TextRank for Keyword Extraction”. *IEEE Access*, 8, 178849–178858, 2020.

[30] Brin S, Page L. “The anatomy of a large-scale hypertextual Web search engine”. *Computer Networks and ISDN Systems*, 30(1–7), 107–117, 1998.

[31] Mihalcea R, Tarau P. "Textrank: Bringing order into text". *Proceedings of the 2004 conference on empirical methods in natural language processing*, Barcelona, Spain, 25-26 July 2004.

[32] Wan X, Xiao J. "Single document keyphrase extraction using neighborhood knowledge". *Proceedings of the 23rd National Conference on Artificial Intelligence*, Chicago/Illinois, USA, 13 – 17 July 2008.
[33] Rose S, Engel D, Cramer N, Cowley W. "*Automatic Keyword Extraction from Individual Documents*". Editors: Berry MW, Kogan J. Text Mining: Applications and Theory, 1–20, Hoboken/New Jersey, USA, John Wiley & Sons, Ltd, 2010.
[34] Danesh S, Sumner T, Martin JH. "Sgrank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction". *Proceedings of the fourth joint conference on lexical and computational semantics*, Denver/Colorado, USA, 4–5 June 2015.
[35] Florescu C, Caragea C. "PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents". *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, 30 July–4 August 2017.
[36] Sarracén GLDP, Rosso P. "Offensive keyword extraction based on the attention mechanism of BERT and the eigenvector centrality using a graph representation". *Personal and Ubiquitous Computing*, 27(1), 45-57, 2023.
[37] Gupta A, Chadha A, Tewari V. "A Natural Language Processing Model on BERT and YAKE technique for keyword extraction on sustainability reports". *IEEE Access*, 12, 7942–7951, 2024.
[38] Londhe RA, Nikam MV. "A Survey on Keyword Extraction Approaches". *International Journal of Advance Research and Innovative Ideas in Education*, 3(3), 3549- 3555, 2017.
[39] Beliga S, Meštrović A, Martinčić-Ipšić S. "An Overview of Graph-Based Keyword Extraction Methods and Approaches". *Journal of Information and Organizational Sciences*, 39(1), 1–20, 2015.
[40] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. "Distributed Representations of Words and Phrases and their Compositionality". *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, Lake Tahoe/Nevada, USA, 5-8 December 2013.
[41] Kim SN, Medelyan O, Kan MY, Baldwin T, Pingar LP. "SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles". *Proceedings of the 5th International Workshop on Semantic Evaluation*, Uppsala, Sweden, 15-16 July 2010.
[42] Liuhuanyong. "GitHub - liuhuanyong/TextGrapher: Text Content Grapher based on keyinfo extraction by NLP method 输入一篇文档，将文档进行关键信息提取，进行结构化，并最终组织成图谱组织形式，形成对文章语义信息的图谱化展示".
https://github.com/liuhuanyong/TextGrapher/ (03.02.2025).
[43] Hu J, Li S, Yao Y, Yu L, Yang G, Hu J. "Patent Keyword Extraction Algorithm Based on Distributed Representation for Patent Classification". *Entropy*, 20(2), 104-123, 2018.
[44] Luo L, Zhang L, Peng H. "An unsupervised keyphrase extraction model by incorporating structural and semantic information". *Progress in Artificial Intelligence*, 9(1), 77–83, 2019.
[45] Singh, V., Bolla, B. K. "Hybrid Approach To Unsupervised Keyphrase Extraction". *Procedia Computer Science*, 235, 1498-1511, 2024.
[46] Giarelis, N., Karacapilidis, N. "Deep learning and embeddings-based approaches for keyphrase extraction: a literature review". *Knowledge and Information Systems*, 66(11), 6493-6526, 2024.
[47] Popova, S., Cardiff, J., Danilova, V. "Rapid Unsupervised Keyphrase Extraction from Single Document". *36th Conference of Open Innovations Association (FRUCT)*, 609-616, 2024.