

# Phraseformer: Multimodal Key-phrase Extraction using Transformer and Graph Embedding

N. Nikzad-Khasmakhi<sup>a</sup>, Mohammad-Reza Feizi-Derakhshi<sup>a,\*</sup>, Meysam Asgari-Chenaghlu<sup>a</sup>, M. A. Balafar<sup>b</sup>, Ali-Reza Feizi-Derakhshi<sup>a</sup>, Taymaz Rahkar-Farshi<sup>a,c</sup>, Majid Ramezani<sup>a</sup>, Zoleikha Jahanbakhsh-Nagadeh<sup>a,d</sup>, Elnaz Zafarani-Moattar<sup>a,e</sup>, Mehrdad Ranjbar-Khadivi<sup>a,f</sup>

<sup>a</sup>Computerized Intelligence Systems Laboratory, Department of Computer Engineering, University of Tabriz, Tabriz, Iran

<sup>b</sup>Department of Computer Engineering, University of Tabriz, Tabriz, Iran

<sup>c</sup>Department of Software Engineering, Ayvansaray University, Istanbul, Turkey

<sup>d</sup>Department of Computer Engineering, Naghadeh Branch, Islamic Azad University, Naghadeh, Iran

<sup>e</sup>Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

<sup>f</sup>Department of Computer Engineering, Shabestar Branch, Islamic Azad University, Shabestar, Iran.

---

## Abstract

**Background** Keyword extraction is a popular research topic in the field of natural language processing. Keywords are terms that describe the most relevant information in a document. The main problem that researchers are facing is how to efficiently and accurately extract the core keywords from a document. However, previous keyword extraction approaches have utilized the text and graph features, there is the lack of models that can properly learn and combine these features in a best way.

**Methods** In this paper, we develop a multimodal Key-phrase extraction approach, namely *Phraseformer*, using transformer and graph embedding techniques. In Phraseformer, each keyword candidate is presented by a vector which is the concatenation of the text and structure learning representations. Phraseformer takes the advantages of recent researches such as **BERT and ExEm** to preserve both representations. Also, the Phraseformer treats the key-phrase extraction task as **a sequence labeling problem solved using classification task**.

**Results** We analyze the performance of Phraseformer on three datasets including **Inspec, SemEval2010 and SemEval 2017 by F1-score**. Also, we investigate the performance of different classifiers on Phraseformer method over Inspec dataset. Experimental results demonstrate the effectiveness of Phraseformer method over the three datasets used. Additionally, the **Random Forest classifier gain the highest F1-score among all classifiers**.

**Conclusions** Due to the fact that the combination of BERT and ExEm is more meaningful and can better represent the semantic of words. Hence, Phraseformer significantly outperforms single-modality methods.

**Keywords:** Multimodal representation learning, Keyword extraction, Transformer, Graph embedding

---



---

\*Corresponding author

Email addresses: [n.nikzad@tabrizu.ac.ir](mailto:n.nikzad@tabrizu.ac.ir) (N. Nikzad-Khasmakhi), [mfeizi@tabrizu.ac.ir](mailto:mfeizi@tabrizu.ac.ir) (Mohammad-Reza

## 1. Introduction

There is a phenomenal growth in the quantity of text documents available to users on the different social medias. Hence, it is vital for efficient and effective ways to retrieve and summarize all these documents [1]. Keyword or key-phrase extraction is a solution that identifies a set of the terms that conclude the main idea of a document [2]. This process helps readers to quickly perceive the content of a document. Key-phrase extraction methods can be effectively used by many text mining applications such as indexing, visualization, summarization, topic detection and tracking, clustering and classification [3, 4].

A number of studies have been conducted to tackle the key-phrase extraction issue. Some of these approaches are textual models that focus only the content of the document to obtain keywords. A group of one have used lexical and syntactic analyses. Other approaches of this class take the advantages of numerical statistics such as term frequency (TF) or term frequency-inverse document frequency (TF-IDF) [5]. On the other hand, graph-based approaches create a various group of methods by constructing a graph of words. In this class, the most central nodes illustrates keywords [1]. Also, we can find hybrid models that uses the combination of textual and graph-based methods to select keywords. In any case, the topic of which combination method is the most suitable and powerful for catching key-phrases extraction is open for discussion.

In this paper, we aim to propose a combined method of a graph-based model and a textual model. Also, we view the key-phrases extraction task as classification and sequence tagging problems. After pre-processing part that removes the stop words and punctuation mark, we construct an undirected and unweighted co-occurrence graph for all documents in the corpus. Then, we use three graph embedding techniques consisting ExEm, Node2vec and DeepWalk to learn structure representations of words. On the other hand, each single document is injected to BERT Transformer to get the word embedding. In the next step, we concatenate two embeddings obtained from the graph and the content of the document to create a single representation for each word. After that, we formulate the key-phrase extraction as sequence labeling task that assigns the BIO tagging to each word in a document. Also, our proposed method treats the sequence labeling as a classification task. The output of the classification is a BIO tag where a word is at the beginning (B) of a key-phrase, inside (I) of a key-phrase and outside of a key-phrase (O). The major contributions of this paper can be summarized as follows:

- To the best of our knowledge, this work is the first to apply a multimodal approach to extract key-phrases using Transformer and graph embedding techniques.

---

Feizi-Derakhshi ), m.asgari@tabrizu.ac.ir (Meysam Asgari-Chenaghlu), balafarila@tabrizu.ac.ir (M. A. Balafar), derakhshi96@ms.tabrizu.ac.ir (Ali-Reza Feizi-Derakhshi), taymazfarshi@ayvansaray.edu.tr (Taymaz Rahkar-Farshi), m\_ramezani@tabrizu.ac.ir (Majid Ramezani), zoleikha.jahanbakhsh@srbiau.ac.ir (Zoleikha Jahanbakhsh-Nagadeh), e.zafarani@iaut.ac.ir (Elnaz Zafarani-Moattar), mehrdad.khadivi@iaushab.ac.ir (Mehrdad Ranjbar-Khadivi)

- Also, to the extent of our knowledge, this is the primary prospective study that directly uses the graph embedding techniques to convert the words in the co-occurrence graph into low-dimensional vectors and employs these vectors to find key-phrases.
- We consider the problem of key-phrase extraction from a document as a sequence labeling task. Moreover, we observe the sequence labeling in the form a classification task using the BIO encoding scheme.

The rest of the paper is structured as follows: Section 2 reviews the related works. Section 3 presents our proposed method and explains it in detail. The descriptions of our experiments are presented in Section 4. Section 5 provides the experimental results. Finally, Section 6 concludes the paper.

## 2. Related Work

The exiting techniques used for keyword extraction can fall into **three** groups: **textual, graph-based and hybrid models**. Textual approaches generate keywords directly from the original text by applying natural language processing techniques. **While, graph-based methods convert the document into a co-occurrence graph where nodes represent words and edges show the relationship between two words in a context window**. On the other hand, hybrid models take the advantage of both text and graph representations of a document to detect keywords. In the following paragraphs, we will investigate these three categories in more detail. In general, the main contributions of our research are summarized as below.

**In textual model the aim is to generate keywords directly from the original text** [5]. The simplest model in this category uses TF-IDF technique to extract keywords. After that, researches have focused on machine learning approaches to train a classifier to capture keywords. With the advent of deep learning approaches such Convolutional Neural Networks (CNNs), Recurrent neural network (RNN) architectures such as Long short-term memory (LSTM), and today’s Transformers have been popular solutions to this task. There are a number of textual approaches including KEA [6], KP-Miner [7], WINGNUS [8], RAKE [9], YAKE [10], TNT-KID [11], [12, 13, 14, 15, 16].

The main idea of graph-based methods is to construct a co-occurrence graph form documents. The co-occurrence network shows the interactions of words in a corpus. In this graph, words represent nodes and there is an edge between two words if these words co-occur within a window. After constructing the co-occurrence graph, some centrality measures such as degree, closeness, betweenness and eigenvector are applied on it to find keyphrases. In these methods, **the keywords are identified by the most central nodes**. A number of methods include TextRank [17], CollabRank [18], DegExt[19], NE-Rank [20], TopicRank [21], Positionrank [22], M-GCKE [5], [23, 24, 25, 26, 27, 1] that use the graph theory to select keywords.

The hybrid models attempt to join two previous mentioned categories. These models calculate scores for words from both co-occurrence graph and the document content. Different approaches use various ways to combine these scores. Authors in studies [28, 29, 30, 31] have proposed hybrid approaches.

Studies [32, 33, 34, 35, 36] have conducted reviews of key-phrase and keyword extraction techniques. In our study, by combining the graph-based and text models, and using sequence tagging and classification, we attempt to develop an effective keyword extraction approach that can eliminate the drawbacks of previous studies.

### 3. Proposed Method

In this section, we propose a framework, called Phraseformer, that extracts key-phrases through learning context representations from text information and node representations in co-occurrence network. The overall processing steps of Phraseformer are explained in Figure 1. The flow of framework is composed of **four main steps**, as follows:

**Text learning:** Find a textual representation for all words using BERT Transformer. This part of Phraseformer provides a deeper understanding to evaluate semantic similarity between words.

**Structure learning:** Create context of each word from the co-occurrence graph using graph embedding techniques to learn structure representation.

**Final word representation:** Concatenate the text information and structure information and create a single representation for each word.

**Sequence labeling and classification:** Formulate key-phrase extraction as a sequence labeling task and label each word based on the BIO tagging scheme through a fully connected layer to classify the word.

#### 3.1. Text Learning

The first step of Phraseformer is to generate the textual vector for every word. With the advent of Transformers, the way of working with text data has truly altered. Transformers eliminate the drawbacks of RNN and CNN architectures. Applying self-attention allows Transformers to be much more parallelization than other architectures [37]. A Transformer is composed of encoding and decoding components as shown in Figure 2. An encoding component includes a number of encoder blocks which have two layers: a Multi-Head Attention layer and a Feed Forward Neural Network layer [38]. On the other side, blocks with a Masked Multi-Head Attention layer before the feed forward layer create the decoding component [39]. Also, both components contain the same number of blocks. There are different models based on Transformer structure such as BERT [40], OpenGPT [41, 42], XLNet [43] and ELMo [44]. In this study, the technique to learn textual representation is BERT Transformer whose structure is presented in Figure 3. One of the important advantages of BERT over models like Word2Vec is that BERT creates the word embedding for each word

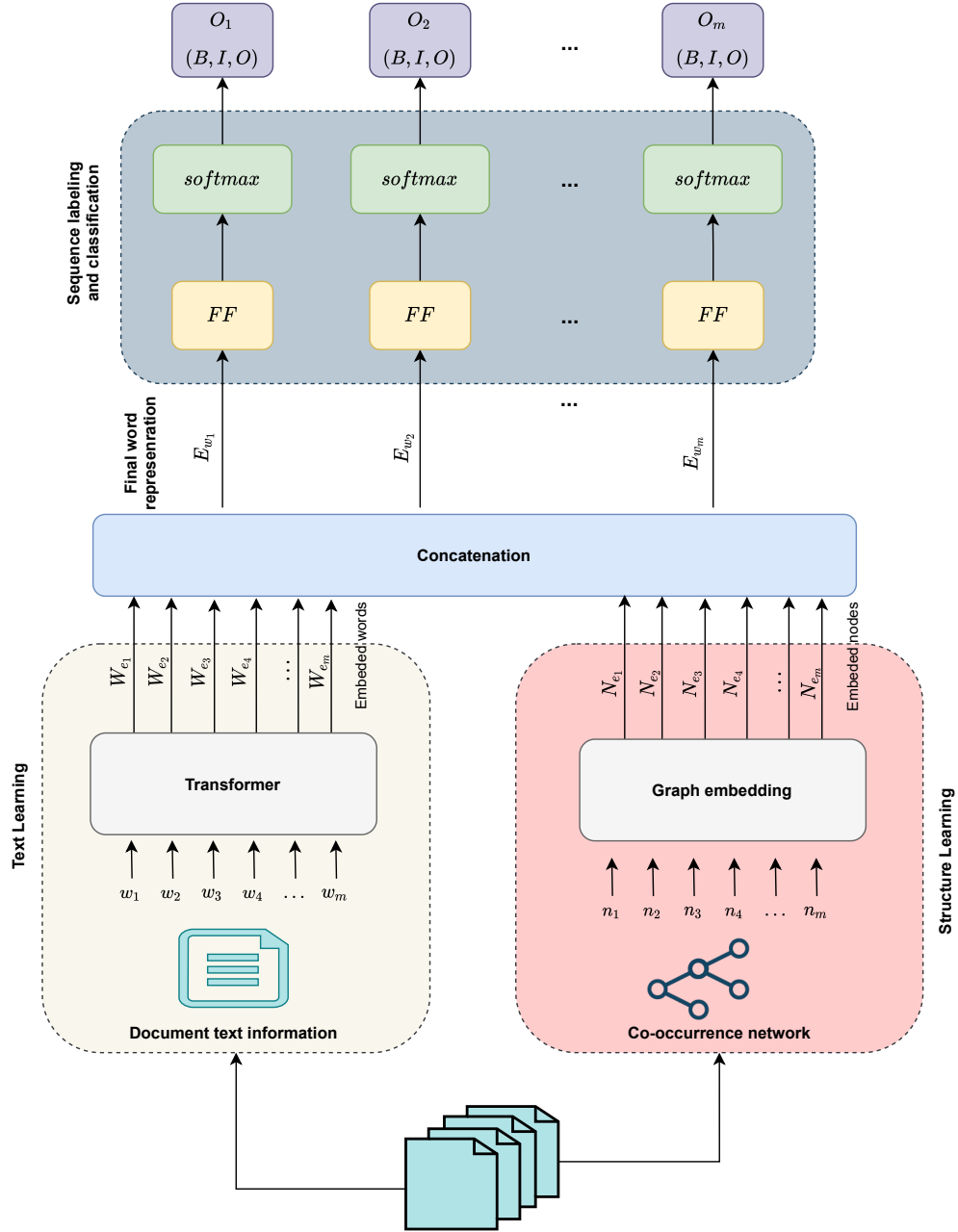


Figure 1: The overall structure of Phraseformer.

based on the each sentence or each document the word is in it. That means that BERT is capable of capturing the context of a word in a document. The text information of a document is composed of the documents' titles, abstracts. The left block in Figure 1 shows the process for learning textual word vectors.

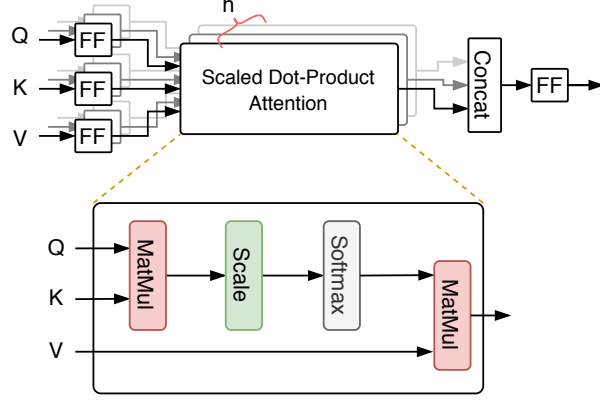


Figure 2: The Transformer model architecture [45].

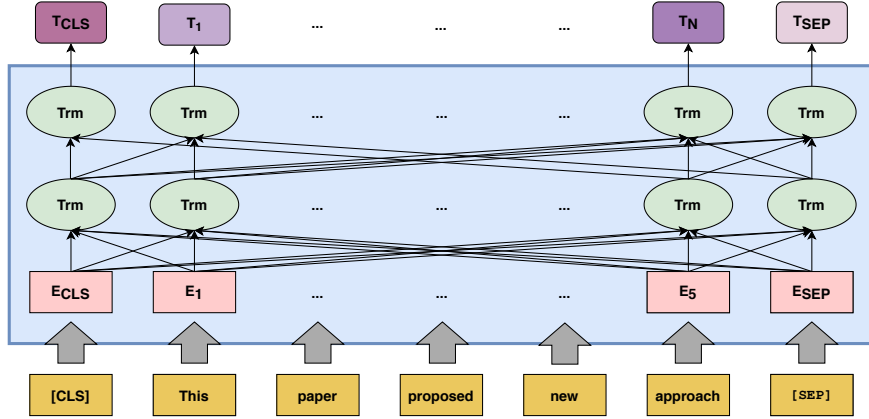


Figure 3: The BERT Transformer architecture [46].

### 3.2. Structure Learning

The second step is to learn the structural vector for every word based on three graph embedding algorithms including ExEm [47], Node2vec [48] and DeepWalk [49] in the co-occurrence network. A co-occurrence network is defined as a graph  $G = (V, E)$  where nodes represent words and each edge  $e \in E$  demonstrates the co-occurrence relationship between word  $v_i$  and word  $v_j$  that appear in a context window. Since the quantity of information that can be acquired from a solitary document is finite, we construct the co-occurrence graph over the all documents instead of a single document, as right block in Figure 1 shown. It should be noted that before building this graph, we remove the stop words and punctuation mark. Then, our task is to learn

the dimensional latent representations of words from this graph using graph embedding techniques. There are different node representation learning algorithms. According to the results reported by different studies, **we select three random walk based graph embedding approaches** that present better performance. In DeepWalk, a set of random walks are generated by starting from every node in the graph. While, Node2vec proposes a biased random walk method that is modified version of DeepWalk. This method uses two parameters to control over the search space. Moreover, ExEm is another technique that uses dominating set theory to create random walks. ExEm characterizes the local neighborhoods by starting each path sampled with a dominating node. Also, the global structure information is captured by selecting another dominating node in the random walk. All three approaches inject these random walks into a skip-gram neural network model to learn node representations.

### 3.3. Final Word Representation

After we obtain each word’s vector representation of text information and co-occurrence network structure, the following step is to combine these information into a single representation. We believe that the concatenation of text-based and structure-based information can better discover the potential of words for being keywords. So, we present each word with a solitary vector which is the combination of text and structure vectors. For example for word  $w_i$ , we show  $E_{w_i} = W_{e_i} + N_{e_i}$  where  $W_{e_i}$  and  $N_{e_i}$  denote the text and structure learning representations for word  $w_i$ , respectively.

### 3.4. Sequence Labeling and Classification

Sequence labeling is a type of pattern recognition task in NLP domain that categorizes words in a text and assigns a class or label to each word in a given input sequence [50, 51]. There are numerous techniques for sequence labeling task include **Hidden Markov Models** [52], **Conditional Random Fields (CRF)** [53] and deep learning approaches.

In this paper, we consider the problem of key-phrase extraction from a document as a sequence labeling task. Also, we observe the sequence labeling in the form a classification task using the BIO encoding scheme as output labels. So, our model takes  $E_{w_1}, E_{w_2}, \dots, E_{w_m}$  as inputs and assigns each word a label  $O_i \in \{B, I, O\}$  where  $B$  shows that  $w_i$  is the beginning of a key-phrase,  $I$  denotes that  $w_i$  is inside a key-phrase, and finally  $O$  illustrates that  $w_i$  is outside of a key-phrase. It can be conducted from Figure 1 that a fully connected structure and a softmax layer are used to make the classification decision.

## 4. Experimental Evaluation

In this section of our research, we will clarify that which datasets have been used to evaluate Phraseformer. Also, we are going to describe the baseline algorithms to compare our proposed method against them. In the next part, we will present different versions of Phraseformer. Finally, evaluation metrics will be specified.

#### 4.1. Dataset

To assess the efficiency of Phraseformer, we used three datasets including Inspec [54], SE-2010 [55] and SE-2017 [56]. We will describe these datasets in the succeeding paragraphs.

**Inspec** includes **abstracts** of papers from Computer Science collected between the years 1998 and 2002.

**SE-2010** contains of full scientific articles that are obtained from the ACM Digital Library. In our experiment, we **used the abstract** of papers.

**SE-2017** consists of **paragraphs** selected from 500 ScienceDirect journal papers from Computer Science, Material Sciences and Physics domains.

It should be noted that because of formulating the key-phrase extraction as a sequence labeling task, we take into consideration key-phrases that appear in the abstracts of articles in three datasets. Table 1 shows the statistics<sup>1</sup>. of the above three datasets.

Table 1: Dataset statistics.

Dataset	Type of doc	#Doc	#Gold Keys (per doc)	#Tokens per doc	Absent Gold Key
Inspec	Abstract	2000	29230 (14.62)	128.20	37.7%
SemEval2010	Paper	243	4002 (16.47)	8332.34	11.3%
SemEval2017	Paragraph	493	8969 (18.19)	178.22	0.0%

#### 4.2. Baseline models

In this study, we compared Phraseformer method with four graph-based approaches and three textual methods. In the following paragraphs, we will explain these techniques in more detail.

**TextRank** [17] is the simplest graph-based method that is based on PageRank algorithm.

**DeepWalk** [49] is a graph embedding technique based on random walks. This method represents each node as a low-dimensional vector.

**Node2vec** [48] is modified version of DeepWalk that uses a biased random walks to convert nodes into vectors.

**ExEm** [47] is a random walk based approach that uses dominating set theory to generate random walks.  $\text{ExEm}_{w2v}$  and  $\text{ExEm}_{ft}$  are two different version of ExEm.

**Word2Vec** [57] learns the vector representations of words. This method passes the document through two feed-forward layers to created vectors [37].

**BiLSTM-CRF** [58] is a textual method that considers the keyphrase extraction as a sequence labeling task using a BiLSTM-CRF architecture.

---

<sup>1</sup>We obtained this information from <https://github.com/LIAAD/KeywordExtractor-Datasets>



**BERT** [40] is a textual approach that uses the transformer structure to obtain the document representation.

#### 4.3. Method Variations

We present four variations of Phraseformer that employ different graph embedding techniques for structure learning. Phraseformer(BERT, DeepWalk), Phraseformer(BERT, Node2vec), Phraseformer(BERT, ExEm<sub>w2v</sub>) and Phraseformer(BERT, ExEm<sub>ft</sub>) are different models of Phraseformer that use DeepWalk, Node2vec, ExEm<sub>w2v</sub> and ExEm<sub>ft</sub> approaches to obtain word representation from the co-occurrence graph, respectively.

#### 4.4. Metrics

To measure the experimental effect, we used F1-score that is formulated as follow:

$$F1 - score = 2 \times \frac{\frac{Y \cap Y'}{Y'} \times \frac{Y \cap Y'}{Y}}{\frac{Y \cap Y'}{Y'} + \frac{Y \cap Y'}{Y}} \quad (1)$$

here  $Y'$  and  $Y$  are the predicted keywords and the real keywords, respectively.

### 5. Experimental Results

This section presents the experimental results. Firstly, we compare Phraseformer against of baselines. Then, we test different classifiers for the classification part and report consequences.

#### 5.1. Baseline comparisons

we compare our multimodal model against textual and graph-based methods. Table 2 presents the results. As expected, our model significantly outperforms all the methods. From the results we have the following observations: **i)** It can be concluded that Phraseformer approaches gain the highest F1-scores. Phraseformer(BERT, ExEm<sub>ft</sub>) strengthens the performance by 6.4%, 19.94% and 13.70% compared with BERT over Inspec, SE-2010 and SE-2017 datasets, respectively. Moreover, Phraseformer(BERT, ExEm<sub>ft</sub>) outperforms ExEm by 112.6%, 290.4% and 130.4% over the mentioned datasets. These high scores prove that our hypothesis about using multimodal learning, sequence labeling and classification is true. **ii)** It is evident that two graph-based and textual methods indicate poor performances. **iii)** It is obvious that textual methods have satisfactory outcomes compared with graph-based models. **iv)** Moreover, BERT shows better results than other textual methods. Because BERT embeds words into vectors by considering the meaning of the words in the sentence. **v)** Additionally, the results of graph-based methods demonstrate that the graph embedding techniques gain the highest F1 scores than TextRank. Also, various versions of ExEm are more successful than other graph embedding approaches. The reason is that ExEm obeys the homophily and structural role equivalences in learning node representations with help of dominating nodes.

Table 2: Comparison with baseline methods (F1-score).

Category	Model	Dataset		
		Inspec	SE-2010	SE-2017
Graph-based method	TextRank [17]	0.1780	0.1990	0.2090
	DeepWalk [49]	0.3190	0.1102	0.2887
	Node2vec [48]	0.3138	0.1098	0.2863
	ExEm <sub>w2v</sub> [47]	0.3273	0.1233	0.2911
	ExEm <sub>ft</sub> [47]	0.3286	0.1246	0.2913
Text-based method	Word2Vec [57]	0.4730	0.2080	0.2920
	BiLSTM-CRF [58]	0.5930	0.3570	0.5210
	BERT [40]	0.6564	0.4056	0.5904
Hybrid method	Phraseformer(BERT, Deep-Walk)	0.6844	0.4722	0.6570
	Phraseformer(BERT, Node2vec)	0.6868	0.4746	0.6594
	Phraseformer(BERT, ExEm <sub>w2v</sub> )	0.6970	0.4848	0.6696
	Phraseformer(BERT, ExEm <sub>ft</sub> )	<b>0.6987</b>	<b>0.4865</b>	<b>0.6713</b>

### 5.2. Classifier

In this part of our experiment we aim to investigate which classifier is best suited for sequence labelling and classification tasks to find key-phrases. Table 3 illustrates the performance of different classifiers on Phraseformer method over Inspec dataset. From the results, it can be seen that Random Forest classifier is obtained the best performance among all classifiers, as shown by the underlined result. On the other hand, the bold results in this table present which models of Phraseformer can gain the highest values on each classifier. It is obvious that the combination of BERT and ExEm is more meaningful and can better represent the semantic of words. Hence, Phraseformer can accurately extract keywords.

Table 3: Classifiers comparison on Inspec dataset (F1-score)

Model	Classifier			
	Random Forest	SVM	Logistic regression	Fully connected
Phraseformer(BERT, Deep-Walk)	0.7024	0.6564	0.6714	0.6844
Phraseformer(BERT, Node2vec)	0.7048	0.6588	0.6738	0.6868
Phraseformer(BERT, ExEm <sub>w2v</sub> )	0.7150	<b>0.6707</b>	<b>0.6857</b>	0.6970
Phraseformer(BERT, ExEm <sub>ft</sub> )	<b><u>0.7167</u></b>	0.6690	0.6840	<b>0.6987</b>

## 6. Conclusion

In this paper, a multimodal approach, Phraseformer, to extract key-phrases from documents is proposed. In this approach, two modalities originate from the co-occurrence graph and the content of documents. Moreover, we formulate the keyphrase extraction as a sequence labeling task solved using a classification model. To represent information of the modalities as vectors, we use BERT Transformer and graph embedding techniques. Utilization of graph based information with aid of textual semantics provide a more insightful representation of the keywords that yield into a more robust keyphrase extraction method. Finally, to validate the effectiveness of the proposed Phraseformer approach, we conduct experiments on three datasets, and the results demonstrate how Phraseformer significantly outperforms single-modality methods.

## Acknowledgment

This project is supported by a research grant of the University of Tabriz (Number S/806).

## Declarations

### *Funding*

This project is supported by a research grant of the University of Tabriz (Number S/806).

### *Conflicts of interests*

The authors have no conflicts of interest to declare that are relevant to the content of this article.

## *Ethical approval*

This article does not contain any studies with human participants or animals performed by any of the authors.

## References

- [1] D. A. Vega-Oliveros, P. S. Gomes, E. E. Milios, L. Berton, A multi-centrality index for graph-based keyword extraction, *Information Processing & Management* 56 (6) (2019) 102063. doi:10.1016/j.ipm.2019.102063.
- [2] M. W. Berry, J. Kogan, Text mining: applications and theory, John Wiley & Sons, 2010.
- [3] S. Lahiri, Keywords at work: Investigating keyword extraction in social media applications, Ph.D. thesis (2018).
- [4] C. Zhang, Automatic keyword extraction from documents using conditional random fields, *Journal of Computational Information Systems* 4 (3) (2008) 1169–1180.
- [5] B. Wang, B. Yang, S. Shan, H. Chen, Detecting hot topics from academic big data, *IEEE Access* 7 (2019) 185916–185927. doi:10.1109/ACCESS.2019.2960285.
- [6] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, C. G. Nevill-Manning, Kea: Practical automated keyphrase extraction, in: *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, IGI global, 2005, pp. 129–152. doi:10.4018/978-1-59140-441-5.ch008.
- [7] S. R. El-Beltagy, A. Rafea, Kp-miner: A keyphrase extraction system for english and arabic documents, *Information systems* 34 (1) (2009) 132–144. doi:10.1016/j.is.2008.05.002.
- [8] T. D. Nguyen, M.-T. Luong, Wingnus: Keyphrase extraction utilizing document logical structure, in: *Proceedings of the 5th international workshop on semantic evaluation*, 2010, pp. 166–169. URL <https://www.aclweb.org/anthology/S10-1035>
- [9] S. Rose, D. Engel, N. Cramer, W. Cowley, Automatic keyword extraction from individual documents, *Text mining: applications and theory* 1 (2010) 1–20. doi:10.1002/9780470689646.ch1.
- [10] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, A. Jatowt, Yake! keyword extraction from single documents using multiple local features, *Information Sciences* 509 (2020) 257–289. doi:10.1016/j.ins.2019.09.013.
- [11] M. Martinc, B. Škrlić, S. Pollak, Tnt-kid: Transformer-based neural tagger for keyword identification, *arXiv preprint arXiv:2003.09166* (2020).
- [12] M. Basaldella, E. Antolli, G. Serra, C. Tasso, Bidirectional lstm recurrent neural network for keyphrase extraction, in: *Italian Research Conference on Digital Libraries*, Springer, 2018, pp. 180–187. doi:10.1007/978-3-319-73165-0\_18.
- [13] R. Alzaidy, C. Caragea, C. L. Giles, Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents, in: *The world wide web conference*, 2019, pp. 2551–2557. doi:10.1145/3308558.3313642.
- [14] M. Tang, P. Gandhi, M. A. Kabir, C. Zou, J. Blakey, X. Luo, Progress notes classification and keyword extraction using attention-based deep learning models with bert, *arXiv preprint arXiv:1910.05786* (2019).
- [15] J. Wang, F. Song, K. Walia, J. Farber, R. Dara, Using convolutional neural networks to extract keywords and keyphrases: A case study for foodborne illnesses, in: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, IEEE, 2019, pp. 1398–1403. doi:10.1109/ICMLA.2019.00228.
- [16] Y. Kim, J. H. Lee, S. Choi, J. M. Lee, J.-H. Kim, J. Seok, H. J. Joo, Validation of deep learning natural language processing algorithm for keyword extraction from pathology reports in electronic health records, *Scientific Reports* 10 (1) (2020) 1–9. doi:10.1038/s41598-020-77258-w.
- [17] R. Mihalcea, P. Tarau, TextRank: Bringing order into text, in: *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004, pp. 404–411. URL <https://www.aclweb.org/anthology/W04-3252>

- [18] X. Wan, J. Xiao, Collabrank: towards a collaborative approach to single-document keyphrase extraction, in: Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), 2008, pp. 969–976.  
URL <https://www.aclweb.org/anthology/C08-1122>
- [19] M. Litvak, M. Last, H. Aizenman, I. Gobits, A. Kandel, Degext—a language-independent graph-based keyphrase extractor, in: Advances in intelligent web mastering-3, Springer, 2011, pp. 121–130. doi:10.1007/978-3-642-18029-3\_13.
- [20] A. Bellaachia, M. Al-Dhelaan, Ne-rank: A novel graph-based keyphrase extraction in twitter, in: 2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Vol. 1, IEEE, 2012, pp. 372–379. doi:10.1109/WI-IAT.2012.82.
- [21] A. Bougouin, F. Boudin, B. Daille, TopicRank: Graph-based topic ranking for keyphrase extraction, in: Proceedings of the Sixth International Joint Conference on Natural Language Processing, Asian Federation of Natural Language Processing, Nagoya, Japan, 2013, pp. 543–551.  
URL <https://www.aclweb.org/anthology/I13-1062>
- [22] C. Florescu, C. Caragea, Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1105–1115. doi:10.18653/v1/P17-1102.
- [23] F. Boudin, A comparison of centrality measures for graph-based keyphrase extraction, in: Proceedings of the sixth international joint conference on natural language processing, 2013, pp. 834–838.  
URL <https://www.aclweb.org/anthology/I13-1102>
- [24] W. D. Abilhoa, L. N. De Castro, A keyword extraction method from twitter messages represented as graphs, Applied Mathematics and Computation 240 (2014) 308–325. doi:10.1016/j.amc.2014.04.090.
- [25] A. Tixier, F. Malliaros, M. Vazirgiannis, A graph degeneracy-based approach to keyword extraction, in: Proceedings of the 2016 conference on empirical methods in natural language processing, 2016, pp. 1860–1870. doi:10.18653/v1/D16-1191.
- [26] M. S. El Bazzl, D. Mammass, T. Zaki, A. Ennaji, A graph-based ranking model for automatic keyphrases extraction from arabic documents, in: Industrial conference on data mining, Springer, 2017, pp. 313–322. doi:10.1007/978-3-319-62701-4\_25.
- [27] F. Boudin, Unsupervised keyphrase extraction with multipartite graphs, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, p. 667–672. doi:10.18653/v1/N18-2105.
- [28] S. Danesh, T. Sumner, J. H. Martin, Sgrank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction, in: Proceedings of the fourth joint conference on lexical and computational semantics, 2015, pp. 117–126. doi:10.18653/v1/S15-1013.
- [29] Y. Zhang, Y. Chang, X. Liu, S. D. Gollapalli, X. Li, C. Xiao, Mike: keyphrase extraction by integrating multidimensional information, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1349–1358. doi:10.1145/3132847.3132956.
- [30] D. Mahata, J. Kuriakose, R. Shah, R. Zimmermann, Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018, pp. 634–639. doi:10.18653/v1/N18-2100.
- [31] D. Mahata, R. R. Shah, J. Kuriakose, R. Zimmermann, J. R. Talburt, Theme-weighted ranking of keywords from text documents using phrase embeddings, in: 2018 IEEE conference on multimedia information processing and retrieval (MIPR), IEEE, 2018, pp. 184–189. doi:10.1109/MIPR.2018.00041.
- [32] S. Siddiqi, A. Sharan, Keyword and keyphrase extraction techniques: a literature review, International Journal of Computer Applications 109 (2) (2015).

- [33] Z. A. Merrouni, B. Frikh, B. Ouhbi, Automatic keyphrase extraction: a survey and trends, *Journal of Intelligent Information Systems* (2019) 1–34 doi:10.1007/s10844-019-00558-9.
- [34] Ö. Ünlü, A. Çetin, A survey on keyword and key phrase extraction with deep learning, in: 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), IEEE, 2019, pp. 1–6. doi:10.1109/ISMSIT.2019.8932811.
- [35] E. Papagiannopoulou, G. Tsoumakas, A review of keyphrase extraction, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10 (2) (2020) e1339. doi:10.1002/widm.1339.
- [36] N. Firoozeh, A. Nazarenko, F. Alizon, B. Daille, Keyword extraction: Issues and methods, *Natural Language Engineering* 26 (3) (2020) 259–291. doi:10.1017/S1351324919000457.
- [37] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, J. Gao, Deep learning based text classification: A comprehensive review, *arXiv preprint arXiv:2004.03705* (2020).
- [38] M. Asgari-Chenaghlu, M.-R. Feizi-Derakhshi, M.-A. Balafar, C. Motamed, et al., Topicbert: A transformer transfer learning based memory-graph approach for multimodal streaming social media topic detection, *arXiv preprint arXiv:2008.06877* (2020).
- [39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, *arXiv preprint arXiv:1706.03762* (2017).
- [40] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
- [41] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).  
URL <https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf>
- [42] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, *OpenAI blog* 1 (8) (2019) 9.
- [43] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, *arXiv preprint arXiv:1906.08237* (2019).
- [44] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, *arXiv preprint arXiv:1802.05365* (2018).
- [45] M. Asgari-Chenaghlu, M. R. Feizi-Derakhshi, L. Farzinvas, C. Motamed, A multimodal deep learning approach for named entity recognition from social media, *arXiv preprint arXiv:2001.06888* (2020).
- [46] N. Nikzad-Khasmakhi, M. Balafar, M. R. Feizi-Derakhshi, C. Motamed, Berters: Multimodal representation learning for expert recommendation system with transformer, *arXiv preprint arXiv:2007.07229* (2020).
- [47] N. Nikzad-Khasmakhi, M. Balafar, M. R. Feizi-Derakhshi, C. Motamed, Exem: Expert embedding using dominating set theory with deep learning approaches, *arXiv preprint arXiv:2001.08503* (2020).
- [48] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. arXiv:1607.00653, doi:10.1145/2939672.2939754.
- [49] B. Perozzi, R. Al-Rfou, S. Skiena, DeepWalk: Online learning of social representations, in: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014. arXiv:1403.6652, doi:10.1145/2623330.2623732.
- [50] Z. He, Z. Wang, W. Wei, S. Feng, X. Mao, S. Jiang, A survey on recent advances in sequence labeling from deep learning models, *arXiv preprint arXiv:2011.06727* (2020).
- [51] A. Akhundov, D. Trautmann, G. Groh, Sequence labeling: A practical approach, *arXiv preprint arXiv:1808.03926* (2018).
- [52] J. Kupiec, Robust part-of-speech tagging using a hidden markov model, *Computer speech & language* 6 (3) (1992) 225–242. doi:10.1016/0885-2308(92)90019-Z.

- [53] J. Lafferty, A. McCallum, F. C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: 18th International Conference on Machine Learning 2001 (ICML 2001), 2001, p. 282–289. doi:10.5555/645530.655813.
- [54] A. Hulth, Improved automatic keyword extraction given more linguistic knowledge, in: Proceedings of the 2003 conference on Empirical methods in natural language processing, 2003, pp. 216–223. doi:10.3115/1119355.1119383.
- [55] S. N. Kim, O. Medelyan, M.-Y. Kan, T. Baldwin, Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles, in: Proceedings of the 5th International Workshop on Semantic Evaluation, 2010, pp. 21–26.
- [56] I. Augenstein, M. Das, S. Riedel, L. Vikraman, A. McCallum, Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications, in: 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, p. 546–555. doi:10.18653/v1/S17-2091.
- [57] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).
- [58] D. Sahrawat, D. Mahata, M. Kulkarni, H. Zhang, R. Gosangi, A. Stent, A. Sharma, Y. Kumar, R. R. Shah, R. Zimmermann, Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings, arXiv preprint arXiv:1910.08840 (2019).