Information Processing and Management 60 (2023) 103356

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/ipm

# Improving unsupervised keyphrase extraction by modeling hierarchical multi-granularity features

Zhihao Zhang [a], Xinnian Liang [b], Yuan Zuo [a], Chenghua Lin [c,*]

[a] School of Economics and Management, Beihang University, Beijing, China
[b] State Key Lab of Software Development Environment, Beihang University, Beijing, China
[c] Department of Computer Science, The University of Sheffield, Sheffield, UK

ARTICLE INFO

*Keywords:*
Unsupervised keyphrase extraction
Graph-based ranking algorithm
Hierarchical Multi-granularity features

ABSTRACT

Existing unsupervised keyphrase extraction methods typically emphasize the importance of the candidate keyphrase itself, ignoring other important factors such as the influence of uninformative sentences. We hypothesize that the salient sentences of a document are particularly important as they are most likely to contain keyphrases, especially for long documents. To our knowledge, our work is the first attempt to exploit sentence salience for unsupervised keyphrase extraction by modeling hierarchical multi-granularity features. Specifically, we propose a novel position-aware graph-based unsupervised keyphrase extraction model, which includes two model variants. The pipeline model first extracts salient sentences from the document, followed by keyphrase extraction from the extracted salient sentences. In contrast to the pipeline model which models multi-granularity features in a two-stage paradigm, the joint model accounts for both sentence and phrase representations of the source document simultaneously via hierarchical graphs. Concretely, the sentence nodes are introduced as an inductive bias, injecting sentence-level information for determining the importance of candidate keyphrases. We compare our model against strong baselines on three benchmark datasets including Inspec, DUC 2001, and SemEval 2010. Experimental results show that the simple pipeline-based approach achieves promising results, indicating that keyphrase extraction task benefits from the salient sentence extraction task. The joint model, which mitigates the potential accumulated error of the pipeline model, gives the best performance and achieves new state-of-the-art results while generalizing better on data from different domains and with different lengths. In particular, for the SemEval 2010 dataset consisting of long documents, our joint model outperforms the strongest baseline UKERank by 3.48%, 3.69% and 4.84% in terms of F1@5, F1@10 and F1@15, respectively. We also conduct qualitative experiments to validate the effectiveness of our model components.

## 1. Introduction

Keyphrase extraction (KE) is the task of extracting from a document a set of salient words or phrases that can summarize the main contents of the document (Hasan & Ng, 2014). Keyphrases have immense value to various downstream text mining tasks, such as text summarization (Song, Huang, & Ruan, 2019), entity recognition and detection (Peng et al., 2021; Zhao, Yan, Cao, & Li, 2021), and question generation (Cheng et al., 2021), to name a few. Broadly speaking, KE methods could be classified into supervised and unsupervised learning methods. Supervised methods require large amounts of labeled training data, which often have limited

* Corresponding author.
*E-mail address:* c.lin@sheffield.ac.uk (C. Lin).

https://doi.org/10.1016/j.ipm.2023.103356
Received 21 July 2022; Received in revised form 14 January 2023; Accepted 13 March 2023
Available online 3 April 2023

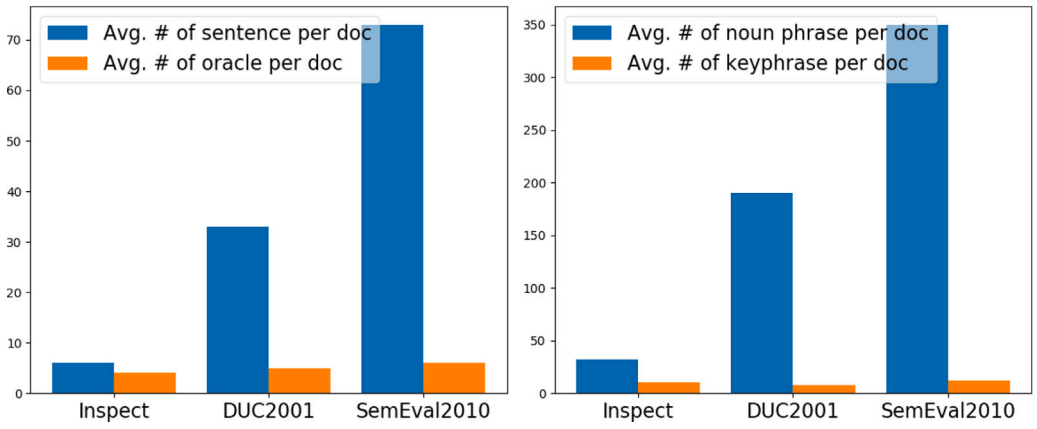


**Fig. 1.** Salience Statistics. Oracle is the salient sentence which contains keyphrases of the document. Noun phrase is the candidate keyphrase which consists of zero or more adjectives followed by one or multiple nouns. Both oracle and keyphrase are salient information of the document which we called salience in this paper.

performance when applied to different domains or dataset types. In contrast to supervised methods, unsupervised methods are more flexible and adaptable by extracting keyphrases based on intrinsic information from the input documents themselves. In this paper, we focus on unsupervised keyphrase extraction (UKE) methods.

UKE methods have been extensively studied in the field of keyphrase extraction. Recently, with the rapid development of representation learning, embedding-based methods (Bennani-Smires, Musat, Hossmann, Baeriswyl, & Jaggi, 2018; Sun, Qiu, Zheng, Wang, & Zhang, 2020) have achieved encouraging performance and become state-of-the-art models. Most embedding-based approaches first calculate the representations of candidate keyphrases and the whole document with static embeddings (e.g. GloVe Pennington, Socher, & Manning, 2014, Sen2Vec Pagliardini, Gupta, & Jaggi, 2018 and Doc2Vec Le & Mikolov, 2014) or contextual representations from pre-trained language models (e.g. BERT Devlin, Chang, Lee, & Toutanova, 2019). Next, candidate keyphrases are ranked by calculating their similarities to the whole document in the vector space. Although these approaches outperform statistics-based (Campos et al., 2018) and topic-based (Boudin, 2018) methods, they ignore the local salience of candidate keyphrases due to the simple similarity calculation between phrases and the whole document, thus limiting their performance. To obtain sufficient information from the context, we previously proposed UKERank (Liang, Wu, Li, & Li, 2021b), which allows joint modeling of the local and global context of the input document. Concretely speaking, UKERank calculates the similarity between candidate keyphrase and the entire document as global similarity, and combines it with boundary-aware centrality which is regarded as local salience to extract keyphrases. The promising results of UKERank are mainly attributed to the modeling of local context which is ignored by previous methods.

Although UKERank demonstrates very good results on benchmark datasets, it focuses on the importance of candidate keyphrases themselves, where the salience of sentences is not taken into account. We hypothesize that the salience of sentences is a very important factor when extracting keyphrases from the full document. For instance, Fig. 1 shows the statistics of the averaged number of sentences/oracles and the averaged number of noun phrases/keyphrases per document across three benchmark datasets. It can be observed that the averaged number of both oracles[1] and keyphrases are much lower than those of sentences and noun phrases.[2] In other words, a significant portion of sentences, especially for long documents such as scholarly documents, may not contain keyphrases. Therefore, taking the full-text of the document as the input for keyphrase extraction may include too many insignificant sentences that contain little salient information, thus hindering the performance of the model. To this end, we propose to explicitly characterize the salience of sentences and phrases via a novel graph-based unsupervised keyphrase extraction model. Table 1 shows an example of hierarchical multi-granularity features from the document, to oracle (sentence), and to keyphrase levels. We can observe that keyphrases often appear in only a few sentences of the entire document. Presumably, selecting the salient sentences from the target document and then extracting keyphrases based on them would be more effective.

Motivated by this observation, in this paper, we propose a novel unsupervised keyphrase extraction model by modeling hierarchical multi-granularity features. Our model includes two model variants, a pipeline model (Sentence-then-Keyphrase Ranking Model) and a joint model (Hierarchical Graph Representation Ranking Model). The pipeline model first extracts salient sentences from the full-text document with the proposed position-aware graph-based ranking algorithm, followed by keyphrase extraction from the extracted salient sentences using the same ranking algorithm. In contrast to the pipeline model which models multi-granularity features in a two-stage paradigm, the proposed joint model accounts for both sentence and phrase representations of the source document simultaneously via hierarchical graphs. By doing so, we convert a flat graph into a hierarchical, non-fully-connected

[1] Salient sentence which contains keyphrases of the document are regarded as Oracle in our paper.

[2] A phrase which consists of zero or more adjectives followed by one or multiple nouns is called Noun Phrase. The keyphrases are usually from the top ranked noun phrases.

**Table 1**
An example from Inspect dataset. Words in red represent the keyphrases.

| |
|---|
| **Document** |
| 1. A friction compensator for pneumatic control valves. 2. A procedure that compensates for static friction -LRB- stiction -RRB- in pneumatic control valves is presented. 3. The compensation is obtained by adding pulses to the control signal. 4. The characteristics of the pulses are determined from the control action. 5. The compensator is implemented in industrial controllers and control systems , and the industrial experiences show that the procedure reduces the control error during stick–slip motion significantly compared to standard control without stiction compensation. |
| **Oracle** |
| 1. A friction compensator for pneumatic control valves. 5. The compensator is implemented in industrial controllers and control systems , and the industrial experiences show that the procedure reduces the control error during stick–slip motion significantly compared to standard control without stiction compensation. |
| **Keyphrases** |
| friction compensator; pneumatic control valves; stiction compensation; stick–slip motion; standard control |

graph, which offers two advantages: (1) it can inject sentence-level information for determining the importance of a candidate keyphrase; and (2) it helps avoid the inherent error propagation of the pipeline model.

Experimental results based on three benchmark datasets show that our proposed models achieve the state-of-the-art performance on the keyphrase extraction task, demonstrating that the keyphrase extraction task indeed benefits from extracted salient sentences. We further conducted detailed analysis on the positive relationship between the keyphrase extraction task and the salient sentence extraction task (see Section 4.3), which reveals that salience sentences can retain the most important information of the full document while significantly reduces the input to be processed by the model.

The main contributions of our work are summarized below: (1) To our knowledge, this is the first attempt to exploit the salience of sentences for unsupervised keyphrase extraction by modeling hierarchical multi-granularity features. (2) Our empirical study shows positive synergy between the salient sentence extraction task and the keyphrase extraction task, suggesting that better integration of these two tasks is a promising research direction. This will be a useful insight to practitioners in the field. (3) Our methods consistently outperform all existing competitors across the three datasets, each with different document lengths, covering two different domains. We also add a more detailed discussion of the analysis of various model components.

The rest of our paper is organized as follows. Section 2 illustrates a detailed explanation of our methodology on how to further improve the performance of keyphrase extraction. Section 3 describes our experimental setups, which is followed by our experimental results in Section 4. Section 5 discusses the difference between our model and UKERank to highlight the theoretical and practical implications of our research. Section 6 reviews the related work of this paper. We finally conclude the paper in Section 7.

## 2. Methodology

In this section, we give a detailed description of our models for unsupervised keyphrase extraction. Considering that our joint and pipeline model share the key model components, we first briefly introduce the pipeline model, and then describe the joint model in detail and its key differences to its counterpart.

### *2.1. Pipeline model: Sentence-then-Keyphrase ranking model*

We first propose a pipeline model which employs a two-stage Sentence-then-Keyphrase paradigm. The pipeline model first extracts salient sentences from the corresponding full-text document using our position-aware graph-based ranking algorithm (see Section 2.2.3). Next, we extract keyphrases based on the extracted salient sentences using the same ranking algorithm. The pipeline model serves a few purposes: (1) it can provide further evidence whether the extracted salient sentences manage to capture the keyphrases, thereby improving the performance of keyphrase extraction; (2) it can support the investigation of impacts on the signal-to-noise ratios of the input document, i.e. the percentage of extracted sentences with respect to the total number of sentences of the document in the salient sentence extraction stage. For more detailed analysis, please refer to Section 4.3

### *2.2. Joint model: Hierarchical graph representation ranking model*

The overall framework of our joint model is illustrated in Fig. 2. Formally, given a document $D$, our model learns to extract a set of salient words or phrases that summarize the main contents of the document. Specifically, the joint model consists of three key components, including (1) Candidate keyphrase generation; (2) Sentence and candidate keyphrase representation; and (3) Keyphrase importance ranking. The first component identifies and extracts the candidate keyphrases, followed by the second component which

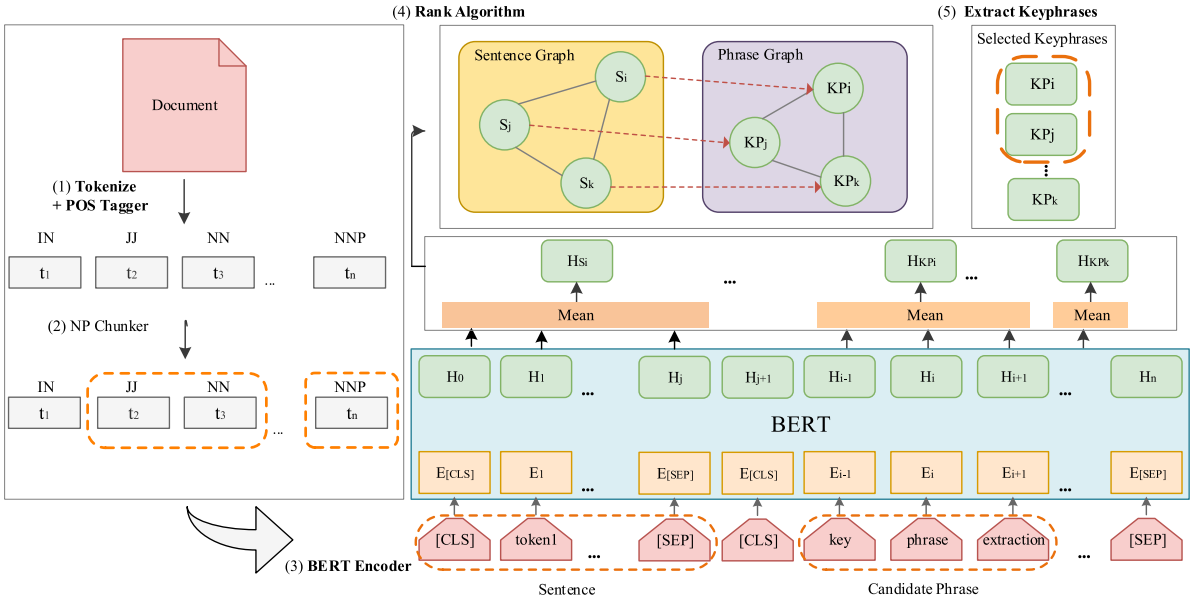


**Fig. 2.** The framework of our hierarchical graph representation ranking model. (1) Tokenize a document into tokens and tag them with POS tags. (2) Identify noun phrases consisting of zero or more adjectives followed by one or more nouns as candidate keyphrase. (3) Generate contextual representations of tokens and sentences with BERT. (4) Score each sentence and candidate keyphrase using our ranking model. (5) Extract keyphrases from the candidate keyphrases according to the scores of the ranking algorithm.

extracts the contextualized representations of sentences and candidate keyphrases. Based on these contextual representations, the last component ranks the importance of keyphrases from two perspectives by simultaneously modeling hierarchical multi-granularity features: phrase-level and sentence-level information. In the following sections, we describe the components in detail.

### 2.2.1. Candidate keyphrase generation

Based on our observation, a keyphrase usually contains more than one word (Sun, Qiu et al., 2020). Therefore, we adopt heuristics to extract noun keyphrases based on their part-of-speech (POS) tag patterns as EmbedRank (Bennani-Smires et al., 2018) does. Specifically, we first tokenize document $D$ into sentences $\{s_1, s_2, \ldots, s_m\}$ and tokens $\{t_1, t_2, \ldots, t_n\}$, and then we use *StanfordCoreNLP Tools*[3] to obtain POS tags of the input text. Finally, we extract noun phrases from the document based on the regular expression which encodes the phrase pattern of interest $\{\langle NN.*|JJ\rangle * \langle NN.*\rangle\}$, where $NN$ and $JJ$ denote the POS tags of noun and adjective, $*$ acts as the wildcard character. This process corresponds to Steps (1) and (2) shown in Fig. 2. The extracted noun phrases are then regarded as candidate keyphrases $\{KP_1, KP_1, \ldots, KP_k\}$ that serve as the input to the subsequent representation methods and ranking algorithms. Note that this part is not the focus of this study and we simply follow this heuristic method as was done for most baselines to make a fair comparison. Nevertheless, we also conducted the further analysis to verify the effectiveness of this common practice (see Section 4.2).

### 2.2.2. Sentence and candidate keyphrases representation

The second component concerns representation learning of candidate keyphrases and input sentences. In contrast to previous work (Bennani-Smires et al., 2018) that uses static vector embedding Sen2Vec (Pagliardini et al., 2018) to represent tokens and sentences, we adopt BERT, a powerful pre-trained language model (PLM), to generate dynamic contextual representations of sentences and candidate keyphrases with Eq. (1). BERT can provide high quality embedding of the text by encoding word, sentence with context dynamically.

$$\{H_1, H_2, \ldots, H_n\} = \mathrm{BERT}(\{t_1, t_2, \ldots, t_n\}), \tag{1}$$

where $H_i$ is the contextualized representation of token $t_i$. The contextualized representation $H_{\mathrm{KP}_i}$ of candidate keyphrases are obtained by averaging the candidate keyphrase's token vector representations. The sentence vector representation $H_{\mathrm{s}_i}$ is derived by averaging the sentence's token vector representations.

In this way, we obtain a set of sentence representations $S = \{H_{\mathrm{s}_i}\}_{i=1,\ldots,m}$ and a set of candidate keyphrase representations $K = \{H_{\mathrm{KP}_i}\}_{i=1,\ldots,k}$. These representations will be used as the input to the core ranking algorithm of our model.

[3] https://stanfordnlp.github.io/CoreNLP/

#### 2.2.3. Keyphrase importance ranking

One of the core technical contributions of our framework is the hierarchical graph representation ranking model. We first briefly describe the traditional graph-based centrality scoring algorithm as the background knowledge. Then we introduce our enhanced position-aware centrality scoring algorithm. Finally, we illustrate the process of characterizing hierarchical multi-granularity features of our ranking model.

**Traditional Graph-based Centrality Scoring.** Given document $D$, a set of sentences $S = \{s_1, s_2, \dots, s_m\}$ and a set of candidate keyphrases $T = \{KP_1, KP_2, \dots, KP_k\}$, a graph-based centrality scoring algorithm treats $D$ as a graph $\mathcal{G} = (V, E)$. Here, $V = \{v_1, v_2, \dots, v_n\}$ is a set of vectors representing the nodes in the graph (e.g. sentences or candidate keyphrases in $D$), and $E = \{e_{ij}\}$ is a $n \times n$ matrix encoding a set of edges, where each element $\{e_{ij}\} \in E$ denotes the weight between vertex $v_i$ and $v_j$. The key idea of traditional graph-based centrality scoring algorithm is to calculate the degree of each node (sentence or candidate keyphrase) as the centrality to measure the importance of the node. A node's centrality can be measured by simply summing all similarities with other nodes. It is computed as follows:

$$\mathbf{C}entrality(v_i) = \sum_{j=1}^{n} e_{ij} \tag{2}$$

Where $e_{ij} = v_i \cdot v_j$ is the dot-product similarity score for each pair $(v_i, v_j)$. Empirically, the simple dot-product usually performs better than other similarity measurements (e.g. cosine similarity) (Zheng & Lapata, 2019). We further conducted the impact of different similarity measure methods (see Section 4.6).

**Position-Aware Graph-based Centrality Scoring.** In general, for most long documents or news articles, the author tends to write the key information at the beginning of the document. Specifically, prior work (Florescu & Caragea, 2017a) point out that the position-biased weight can greatly improve the performance for keyphrase extraction and they employ the sum of the position's inverse of words in the document as the weight. SIFRank (Sun, Qiu et al., 2020) and UKERank (Liang, Wu et al., 2021b) adopt a simpler position bias weight, which only consider where the candidate phrase first appears. To implement this important insight, we also follow this simpler position bias weight as shown in Eq. (4) to make a fair comparison.

$$\mathbf{C}entrality(v_i) = \hat{p}(v_i) \cdot \sum_{j=1}^{n} \max(e_{ij} - \theta, 0) \tag{3}$$

$$\hat{p}(v_i) = \frac{\exp(p(v_i))}{\sum_{k=1}^{n} \exp(p(v_k))} \tag{4}$$

First, a threshold $\theta = \beta \cdot (\max(e_{ij}) - \min(e_{ij}))$ is employed to exclude the effect from the nodes that are far from the current node $i$. It removes their influence on the centrality score by setting all $e_{ij} < \theta$ to zero. $\beta$ is a hyper-parameter that controls the scale. The position bias weight is computed by $p(v_i) = \frac{1}{p_i}$, where $p_i$ is the position of the sentence's (or candidate keyphrase's) first appearance in the document. Finally, the softmax function is adopted to normalize the position bias weight. In this paper, we employ this normalized position-aware centrality to measure the importance of nodes.

**Hierarchical Graph Representation Ranking.** To characterize the features of both salient sentences and keyphrases in a joint manner, our joint model accounts for both sentence and phrase representations of the source document via hierarchical graphs. The two perspectives from phrase-level and sentence-level information are considered to improve the keyphrase importance ranking simultaneously.

Specifically, the sentence nodes are introduced as an inductive bias, injecting sentence-level information for determining the importance of candidate phrases. We use our position-aware graph-based centrality scoring algorithm to compute the centralities of the sentence and candidate keyphrase nodes simultaneously. Then we combine these two centralities as follows:

$$\mathbf{C}entrality(v_i) = \lambda \cdot Centrality(s_i) + Centrality(k_i) \tag{5}$$

where $Centrality(k_i)$ is the centrality score of the candidate keyphrase, and the $Centrality(s_i)$ is the centrality score of the sentence in which the candidate keyphrase is located. $\lambda$ is also a hyper-parameter that controls the weight of the sentence node. We rank candidate keyphrases according to their final score $Centrality(v_i)$, and extract the top $k$ candidate keyphrases as the final keyphrases of the document.

## 3. Experimental setup

### 3.1. Datasets

To fully validate the effectiveness of our model, we evaluate our method on 3 benchmark datasets: **Inspec**, **DUC2001** and **SemEval2010**. Table 2 shows the detailed data statistics. The **Inspec** (Hulth, 2003) consists of 2000 documents that are selected from scientific document abstracts. We choose 500 documents as the testset to verify our method. The **DUC2001** (Wan & Xiao, 2008) is derived from news articles and contains 308 documents. The **SemEval2010** (Kim, Medelyan, Kan, & Baldwin, 2010) is ACM full length paper with keyphrases annotated by authors and readers. We select 100 papers as the testset for our experiment.

**Table 2**
Data statistics. #Doc, #Uni-gram and #N-gram denote the numbers of documents, uni-grams and N-grams. AveWords and AveKeyphrase represent the averaged numbers of words and keyphrases.

| Datasets | Domain | #Doc | AveWords | AveKeyphrase | #Uni-gram | #N-gram |
|---|---|---|---|---|---|---|
| Inspec | Scientific | 500 | 135 | 10 | 662 | 4251 |
| DUC2001 | News | 308 | 846 | 8 | 431 | 2057 |
| SemEval2010 | Scientific | 100 | 1585 | 12 | 235 | 969 |

### 3.2. Baselines and evaluation metrics

We compare our approach with three kinds of models to fully validate the effectiveness of our model. Firstly, we compare our approach with statistics-based models, including TF-IDF and YAKE (Campos et al., 2018). Then, we compare our approach with five popular graph-based models. TextRank (Mihalcea & Tarau, 2004), which converts the document into a graph with co-occurrence of phrases, is the first attempt to rank candidate keyphrases using PageRank algorithm (Page, Brin, Motwani, & Winograd, 1999). SingleRank (Wan & Xiao, 2008) extends TextRank by enhancing the graph construction using a slide window. PositionRank (Florescu & Caragea, 2017b) incorporates position bias in determining the significance of candidate keyphrases. TopicRank (Bougouin, Boudin, & Daille, 2013) mines keyphrases using topic distribution information. MultipartiteRank (Boudin, 2018) ranks candidate keyphrases with graph theory by splitting the whole graph into sub-graphs. Finally, we compare our approach with three strong embedding-based models. Specifically, EmbedRank (Bennani-Smires et al., 2018) measures the similarities between candidate keyphrases and the whole document with Doc2Vec/Sent2Vec embeddings to rank keyphrases. SIFRank (Sun, Qiu et al., 2020) enhances EmbedRank by replacing static embedding with contextual representations from the pre-trained language models. KeyGames (Saxena, Mangal, & Jain, 2020) incorporates game theoretic method into the ranking algorithm. As the strongest baseline, our previous UKERank (Liang, Wu et al., 2021b), which jointly model global and local context to rank candidate keyphrases, is also included.

The performance of keyphrase model is typically evaluated by comparing the top $N$ predicted keyphrases with target keyphrases (ground-truth labels). The evaluation cutoff $N$ can be a fixed number (e.g., F1@5 compares the top-5 keyphrases predicted by the model with the ground-truth to compute an F1 score). We follow the common practice and evaluate the performance of all the models using macro F-measure at the top $N$ keyphrases (F1@N), and $N$ is set to be 5, 10 and 15. Concretely speaking, we report F1@5, F1@10 and F1@15 scores and we apply stemming to both extracted the keyphrases and the ground truth.

### 3.3. Implementation details and hyper-parameter setting

*Stanford CoreNLP Tool* is adopted for tokenizing, part-of-speech tagging, and noun phrase chunking. We employ regular expression $\{\langle NN.*|JJ\rangle * \langle NN.*\rangle\}$ to identify noun phrases as the candidate keyphrases. We use the uncased BERT-base version[4] to generate the contextualized word embeddings. For the pipeline model, we set the number of the extracted salient sentences to 33, 6 and 73, i.e. the averaged numbers of documents, for DUC2001, Inspect, and SemEval2010, respectively. For more details on the impact of this parameter please refer to Section 4.3. For the joint model, we set the hyper-parameter $\lambda$, which is the weight that balances the importance of the sentence node, to 0.2, 0.2 and 1.0 for DUC2001, Inspect, and SemEval2010, respectively. Another hyper-parameter $\beta$, which is used to filter noise, is set to 0.2 for all three datasets. For more details on the sensitivity analysis of these parameters please refer to Section 4.5.

## 4. Experimental results

### 4.1. Overall results

Table 3 shows the keyphrase extraction performance (measured by F1@N) of our approaches and the baselines on three benchmark datasets. It is worth noting that the experimental datasets exhibit distinctive characteristics in terms of averaged document length (cf. Table 2) and domains, and thus serve as a good test bed for assessing the generalizability of the tested models.

In terms of the overall performance, it can be seen that our joint model consistently outperforms all the baselines, achieving the state-of-the-art performance across all three datasets. Our pipeline model gives the second best performance for most of the cases, except on F1@5 and F1@15 for DUC2001. The promising results of our methods are mainly attributed to our modeling of hierarchical multi-granularity features (phrase-level and sentence-level information), which are ignored by previous works. Especially worth mentioning is that our hierarchical graph representation ranking method injects sentence-level information in determining the importance of candidate keyphrase in a joint paradigm. It helps avoid the inherent error propagation of our pipeline model and further the performance of unsupervised keyphrase extraction.

Comparing different baseline methods, embedding-based methods generally give stronger performance than statistics-based and graph-based methods. This is likely attributed to the fact that embedding-based methods learn representations that can better capture the context information of the input document, especially for short documents that contain less aspects of information. We can further

[4] https://github.com/google-research/bert

**Table 3**
Model comparison. The best results are in bold, and the second best underline.

| Models | Inspec | | | DUC2001 | | | SemEval2010 | | |
|---|---|---|---|---|---|---|---|---|---|
| | F1@5 | F1@10 | F1@15 | F1@5 | F1@10 | F1@15 | F1@5 | F1@10 | F1@15 |
| Statistics-based models | | | | | | | | | |
| TF-IDF | 11.28 | 13.88 | 13.83 | 9.21 | 10.63 | 11.06 | 2.81 | 3.48 | 3.91 |
| YAKE | 18.08 | 19.62 | 20.11 | 12.27 | 14.37 | 14.76 | 11.76 | 14.4 | 15.19 |
| Graph-based models | | | | | | | | | |
| TextRank | 27.04 | 25.08 | 36.65 | 11.80 | 18.28 | 20.22 | 3.80 | 5.38 | 7.65 |
| SingleRank | 27.79 | 34.46 | 36.05 | 20.43 | 25.59 | 25.70 | 5.90 | 9.02 | 10.58 |
| TopicRank | 25.38 | 28.46 | 29.49 | 21.56 | 23.12 | 20.87 | 12.12 | 12.90 | 13.54 |
| PositionRank | 28.12 | 32.87 | 33.32 | 23.35 | 28.57 | 28.60 | 9.84 | 13.34 | 14.33 |
| MultipartiteRank | 25.96 | 29.57 | 30.85 | 23.20 | 25.00 | 25.24 | 12.13 | 13.79 | 14.92 |
| Embedding-based Models | | | | | | | | | |
| EmbedRank d2v | 31.51 | 37.94 | 37.96 | 24.02 | 28.12 | 28.82 | 3.02 | 5.08 | 7.23 |
| EmbedRank s2v | 29.88 | 37.09 | 38.40 | 27.16 | 31.85 | 31.52 | 5.40 | 8.91 | 10.06 |
| SIFRank | 29.11 | 38.80 | 39.59 | 24.27 | 27.43 | 27.86 | – | – | – |
| SIFRank+ | 28.49 | 36.77 | 38.82 | 30.88 | 33.37 | 32.24 | – | – | – |
| KeyGames | 32.12 | 40.48 | 40.94 | 24.42 | 28.28 | 29.77 | 11.93 | 14.35 | 14.62 |
| Our model | | | | | | | | | |
| UKERank | 32.61 | 40.17 | 41.09 | 28.62 | 35.52 | 36.29 | 13.02 | 19.35 | 21.72 |
| Pipeline | 34.03 | 40.67 | 41.42 | 28.74 | 35.55 | 36.23 | 14.55 | 21.75 | 24.13 |
| Joint | **34.59** | **40.90** | **42.19** | **31.80** | **36.66** | **38.87** | **18.03** | **25.44** | **28.97** |

**Table 4**
Recall score with different candidate keyphrase generation method.

| Method | Inspec | DUC2001 | SemEval2010 |
|---|---|---|---|
| NPChunk | 76.27 | 87.82 | 73.93 |
| MaxSpan | 15.08 | 19.41 | 41.24 |
| Wiki-Filter | 11.15 | 9.64 | 8.96 |

see that both EmbedRank and SIFRank perform better than graph-based models on short length documents (i.e. DUC2001 and Inspec). KeyGames and our previous work UKERank are more generalized and work well on both short and long input documents. The strength of UKERank is evident on long scientific documents (i.e. SemEval2010). This is mainly attributed to the modeling of local context by boundary-aware centrality (Liang, Wu et al., 2021b).

Although these works above yield good results on benchmark datasets, the existing methods, either graph-based or embedding-based, do not consider the salience of sentences, which inevitably hinder their performance. The large margin gains by our methods (i.e. the pipeline model and joint model) through modeling hierarchical multi-granularity features (i.e. the phrase-level and sentence-level information) strongly demonstrate that our models are very effective. The encouraging results of our pipeline model reveal positive synergy between keyphrase extraction task and salient sentences extraction task. Moreover, the further performance gains by our joint model, which accounts for both sentence and phrase representations of the source document via hierarchical graphs, demonstrate the effectiveness of simultaneously modeling of hierarchical multi-granularity features.

### *4.2. Evaluation of different candidate keyphrase generation method*

To further verify the effectiveness of the heuristic method (noun phrase chunking method, i.e. NPChunk), we select two alternative approaches for candidate keyphrase generation stage. They are MaxSpan (maximum span from all possible spans) and Wiki-Filter (distantly supervised method based on WiKi entities), respectively (Gu et al., 2021). MaxSpan keeps the maximum span as candidate phrase if there exist overlaps among all possible multi-word spans. Wiki-Filter follows a string matching from the Wikipedia entities from all possible multi-word spans. We compare the recall scores of different candidate keyphrase generation methods. As illustrated in Table 4, it can be seen that NPChunk outperforms MaxSpan and Wiki-Filter by a large margin. The average recall score of NPChunk can reach over 79.34, which is sufficient to support subsequent ranking methods.

### *4.3. Impact of the number of extracted sentences*

As discussed in Section 1, only a small fraction of sentences in a document may contain keyphrases. Therefore, the number of extracted salient sentences will have a profound impact on the performance of our pipeline model. We evaluate the performance of our pipeline model with varying numbers of extracted salient sentences. Specifically, we vary the proportions of the number of the extracted salient sentences to the averaged number of sentences, which includes: $\{40, 60, 80, 100\}$ (%). As shown in Fig. 3, there

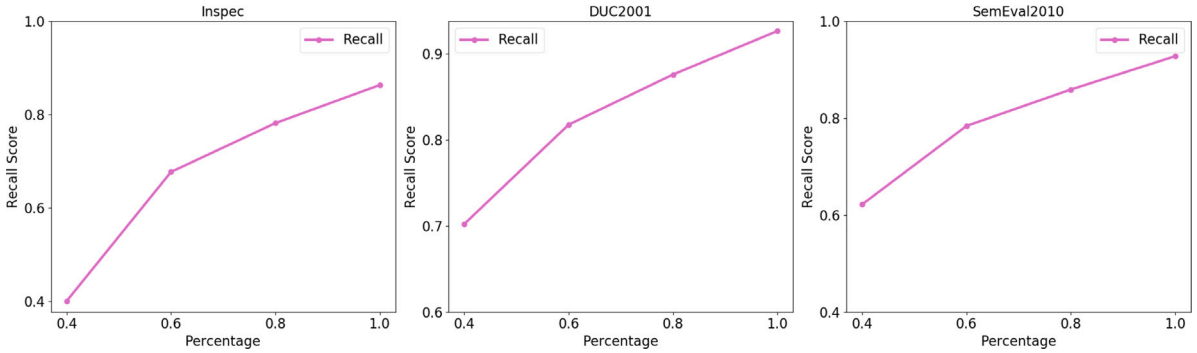


**Fig. 3.** Recall score with different proportions of the number of the extracted sentences to the averaged number of sentences.

**Table 5**
Performance with different proportions of the number of extracted sentences to the averaged number of sentences.

| Models | Inspec | | | DUC2001 | | | SemEval2010 | | |
|---|---|---|---|---|---|---|---|---|---|
| | F1@5 | F1@10 | F1@15 | F1@5 | F1@10 | F1@15 | F1@5 | F1@10 | F1@15 |
| Oracle | **34.88** | **42.91** | **44.05** | **36.14** | **45.21** | **45.33** | **19.75** | **27.30** | **29.96** |
| UKERank | 32.61 | 40.17 | 41.09 | 28.62 | 35.52 | 36.29 | 13.02 | 19.35 | 21.72 |
| Pipeline+40% | 30.57 | 31.49 | 31.33 | 29.18 | 34.64 | 35.15 | 14.63 | 21.75 | 23.97 |
| Pipeline+60% | 33.48 | 38.49 | 37.10 | 28.68 | 35.30 | 35.82 | 14.55 | 21.75 | 24.13 |
| Pipeline+80% | 34.24 | 40.03 | 39.67 | 28.74 | 35.59 | 36.21 | 14.55 | 21.75 | 24.13 |
| Pipeline+100% | 34.03 | 40.67 | 41.42 | 28.74 | 35.55 | 36.23 | 14.55 | 21.75 | 24.13 |

**Table 6**
The results of ablation experiments on three benchmark datasets.

| Models | Inspec | | | DUC2001 | | | SemEval2010 | | |
|---|---|---|---|---|---|---|---|---|---|
| | F1@5 | F1@10 | F1@15 | F1@5 | F1@10 | F1@15 | F1@5 | F1@10 | F1@15 |
| Joint | **34.59** | **40.90** | **42.19** | **31.80** | **36.66** | **38.87** | **18.03** | **25.44** | **28.97** |
| - Sentence | 34.06 | 40.77 | 42.07 | 29.53 | 35.08 | 37.52 | 16.05 | 23.78 | 27.05 |

is a clear trend that the recall scores increase as a higher proportion of the extracted sentences are included. With only 60% of the averaged number of sentences on all datasets, the recall score can reach over 70%.

Similarly, Table 5 presents the detailed performance when the proportions of the extracted sentences are varied on the three datasets. From the results, we can find that our pipeline model even outperforms the strong baseline UKERank with only 60% of the averaged number of sentences in terms of F1@5 on all three datasets. These results indicate that our Sentence-then-Keyphrase ranking model is able to significantly highlight the salient information of the full-text document and improve the performance of unsupervised keyphrase extraction. Note that Oracle baseline can be considered as the upper bound of the pipeline model, which extracts keyphrases from the oracle sentences[5] directly. Presumably, using more recent and state-of-the-art extractive summarization methods will further improve the performance of our pipeline model, which could be investigated in future work.

### *4.4. Impact of sentence node*

We perform ablation study to evaluate the contribution of the sentence node in our joint model, where the results are illustrated in Table 6. It can be observed that our joint model performance degrades after removing sentence nodes. This result verifies the effectiveness of our hierarchical graph representation ranking model. Note that the performance of our model on Inspec is not sensitive to the removal of sentence nodes. One plausible explanation is that the document length of Inspec is very short compared to the other two datasets, the limited number of sentences leads to a limited impact of sentence nodes. However, for relatively long documents, especially for DUC2001 and SemEval2010, the sentence node plays an important role. Overall, we show that modeling hierarchical multi-granularity features in a joint manner is crucial for unsupervised keyphrase extraction.

[5] Salient sentence which contains keyphrases of the document are regarded as Oracle in our paper.

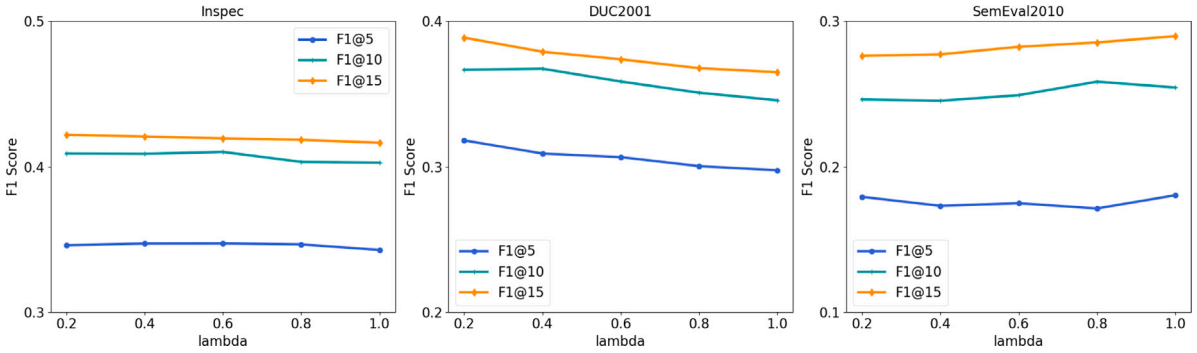


**Fig. 4.** Performance with different $\lambda$.

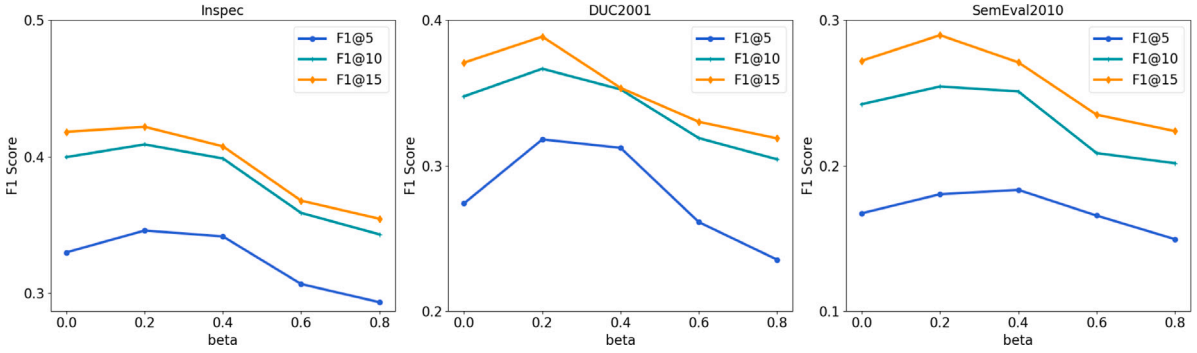


**Fig. 5.** Performance with different $\beta$.

### 4.5. Impact of hyper-parameters

In this section, we analyze the influence of the hyper-parameters and report the optimal settings for each dataset. We have two hyper-parameter $\lambda$ and $\beta$ in our joint model, where $\lambda$ is the weight that balances the importance between the sentence nodes and phrase nodes, $\beta$ is used to control the filter boundary.

We test the performance of our model based on the development set by changing the values of $\lambda$ from 0.0 to 1.0, with an increment step of 0.2. From Fig. 4, the best values of $\lambda$ are 0.2 for DUC2001 and Inspec, and 1.0 for SemEval2010. These settings are intuitive and conform to the characteristics of the experimental datasets. As Inspec comes from the abstracts of scientific papers and DUC2001 is derived from news articles, the number of sentences in these two datasets is relatively small compared to SemEval2010. As a result, salient sentences are more evenly distributed, parameter $\lambda$ should be set to a smaller value to minimize the influence of non-salient sentence nodes. In contrast, SemEval2010 is constructed based on long scientific documents with more unimportant sentences, and hence require setting $\lambda$ with a larger value to maximize the influence of salient sentences nodes.

In our experiment, we follow the setting of prior work (Liang, Wu et al., 2021b) by setting the value of $\beta = 0.2$. To validate the setting, we further evaluate the performance of our model based on the development set by changing the values of $\beta$ from 0.0 to 0.8, with an increment step of 0.2. As shown in Fig. 5, $\beta = 0.2$ is the best setting for all three experimental datasets. Hyper-parameter $\beta$ controls the threshold below which the similarity score is set to 0. The figure shows that hyper-parameter $\beta$ has great impact on the performance of the model. This also shows that noisy phrases/sentences do exist and hurt the performance of centrality-based models.

### 4.6. Impact of different similarity measure method

We adopt dot-product similarity score as centrality scoring in this paper. Empirically, the simple dot-product usually performs better than other similarity measurements (e.g. cosine similarity) for BERT representations (Liang, Li, Wu, Li, & Li, 2021; Liang, Wu, Li, & Li, 2021c; Zheng & Lapata, 2019). To verify it, we also use other similarity measure methods (e.g. cosine similarity). As shown in Table 7, we find that the simple dot-product indeed performs better. One possible explanation is that we use the frozen BERT to encode the text and its representations are only optimized for the masked word prediction task, which does not aim to measure the similarity between words. The normalization of representations may break the learned relative relation between words. This is a very meaningful problem, and we believe investigating this phenomenon could be research problem in itself. We will try to explore this problem in our future work.

**Table 7**
The results of different similarity measure methods for centrality scoring.

| Similarity | Inspec | | | DUC2001 | | | SemEval2010 | | |
|---|---|---|---|---|---|---|---|---|---|
| | F1@5 | F1@10 | F1@15 | F1@5 | F1@10 | F1@15 | F1@5 | F1@10 | F1@15 |
| Dot-product | **34.59** | **40.90** | **42.19** | **31.80** | **36.66** | **38.87** | **18.03** | **25.44** | **28.97** |
| Cosine | 32.79 | 36.69 | 37.24 | 27.79 | 33.02 | 34.79 | 17.15 | 21.98 | 22.45 |

**Table 8**
An example from DUC2001. The extracted salient sentences are underlined. Red text indicates the extracted gold truth. Blue text indicates other phrases extracted by our model.

| |
|---|
| **Article** |
| A formal model of computing with words. Classical automata are formal models of computing with values. Fuzzy automata are generalizations of classical automata where the knowledge about the system 's next state is vague or uncertain. It is worth noting that like classical automata, fuzzy automata can only process strings of input symbols. Therefore, such fuzzy automata are still -LRB- abstract -RRB- devices for computing with values, although a certain vagueness or uncertainty are involved in the process of computation. We introduce a new kind of fuzzy automata whose inputs are instead strings of fuzzy subsets of the input alphabet. These new fuzzy automata may serve as formal models of computing with words. We establish an extension principle from computing with values to computing with words. This principle indicates that computing with words can be implemented with computing with values with the price of a big amount of extra computations. |
| **Gold Truth** |
| formal model; computing with words; fuzzy automata; fuzzy subsets; input alphabet; extension principle; pushdown automata |
| **Our Model** |
| Top5: formal model; words; classic automata; fuzzy automata; input symbols;<br>Top10: fuzzy subsets; values; input alphabet; strings; next state;<br>Top15: knowledge; system; extension principle; generalizations; |

### *4.7. Quantitative analysis*

We further conducted a qualitative analysis (see Table 8) based on DUC2001 to validate the effectiveness of our approach. The top row of Table 8 shows a new article from the dataset, where the extracted salient sentences (i.e., important sentence nodes) are underlined. Phrases in red indicate the extracted gold truth and the phrase in blue denote other phrases extracted by our model. We can see that our model extracts a lot of correct keyphrases which are the same as the ground truth. It is worth mentioning that salient sentences extracted by our model contain the keyphrases successfully. These salient sentences highlight the full-text's important information and shorten the length of the full-text document greatly. By adding more weight to these salient sentence nodes, or just extracting keyphrases based on them, the performance of keyphrase extraction is further improved. This example strongly demonstrates that our modeling of hierarchical multi-granularity features is very effective for unsupervised keyphrase extraction.

## 5. Discussion and implication

Our previously proposed UKERank model[6] (Liang, Wu et al., 2021b) jointly models global and local context. Specifically, for the global view, we compute the similarity between the given phrase and the whole document in the vector space. For the local view, we first build a graph based on the input document itself, where candidate keyphrases are considered as vertices and edges between them are weighted by their similarities. We then propose a boundary-aware centrality computational approach to measure the salience of local context, which is based on the observation that most significant information usually appears at the beginning or end of a document (Dong, Romascanu, & Cheung, 2021). Finally, we combine global and local context modeling for keyphrase ranking. The good experimental results demonstrate that UKERank can effectively capture global and local information. The ablation study also verifies the effectiveness of local context modeling, which was ignored by previous methods.

Nevertheless, UKERank ignores the salience of sentences, which is an important factor for keyphrase extraction. To address this issue, we propose a novel position-aware graph-based unsupervised keyphrase extraction model (pipeline model and joint model) to explicitly characterize phrase-level and sentence-level information simultaneously. The improvement of our models over UKERank validates the effectiveness of our modeling of hierarchical multi-granularity features. Our model also shows positive synergy between the salient sentence extraction task and the keyphrase extraction task, suggesting that better integration of these two tasks is a direction worth studying in depth.

It is worth noting that in addition to modeling the salience of sentences, another difference between UKERank and the proposed models in this paper is that we remove both global similarity and boundary-aware centrality components, and instead adopt a simpler

[6] Note that Xinnian Liang, the second author of this manuscript, is also the first author of UKERank model (Liang, Wu et al., 2021b).

and general position-aware graph-based centrality to score the keyphrases/sentences. Specifically, in terms of global similarity, the representation of a long document may contain multiple aspects of information that affect the similarity between the phrase and the entire document, which results in a limited impact on the global information. In addition, boundary-aware centrality imposes hard constraint on modeling the amount of text from the beginning and the end of the document that are considered important, controlled by a fixed hyper-parameter. In contrast, our algorithm is more flexible in the sense that we add the salience of sentence-level node to the importance of phrase-level node using graph-based centrality scoring where sentences are regarded as vertices and the edges are sentence similarities. Essentially, the sentence-level information can also be regarded a kind of inductive bias, this coarse-grained sentence-level node, which contains more global information compared to fine-grained phrase-level node, helps the model locate salient sentences across the whole document and enhance the representations of the source document to improve the keyphrase importance ranking (it could be regarded as a kind of soft constraint). All of these modifications, including the removal of both global similarity and boundary-aware centrality components, make our model more generic and robust.

## 6. Related work

### 6.1. Pre-trained language model

Pre-trained language models (PLMs) are the type of models that are trained on large amounts of unlabeled corpora through neural networks and then fine-tuned with respect to specific downstream tasks. Different from previous static text representations which are obtained by Word2Vec (Mikolov, Chen, Corrado, & Dean, 2013), GloVe (Pennington et al., 2014), or FastText (Joulin, Grave, Bojanowski, & Mikolov, 2017), PLMs can provide high quality embeddings of the text by encoding words, sentences, or documents with dynamic context. Contextual representations obtained from PLMs can provide better semantic information in contrast with Sen2Vec (Pagliardini et al., 2018) or Doc2Vec (Le & Mikolov, 2014). Moreover, the out of vocabulary (OOV) problem can also be alleviated by calculating text representations dynamically based on the context.

ELMo (Peters et al., 2018) can produce deep contextualized representation by capturing bidirectional information via concatenating the forward and backward layers of the Bi-LSTM model. Recently, autoencoding Pre-trained Language Model BERT (Devlin et al., 2019), a deep bidirectional Transformers model, can capture better contextual representations than ELMo. There are other PLMs such as BERT's variant RoBERTa (Liu et al., 2019), autoregressive XLNet (Yang et al., 2019), Denoising Sequence-to-Sequence BART (Lewis et al., 2020), and Knowledge-enhanced ERNIE 2.0 (Sun, Wang et al., 2020). In this paper, BERT is employed to generate contextual representations of phrases and sentences by pooling their embeddings.

### 6.2. Supervised keyphrase extraction

Typical supervised keyphrase extraction methods consists of two steps: (1) extract candidate keyphrases from the source document and (2) estimate keyphrase importance to rank the candidate keyphrases. In the candidate keyphrase extraction stage, the heuristic algorithms are employed to compose words into n-grams as candidate keyphrases (Gu et al., 2021; Wang, Fan, & Rosé, 2020; Xiong, Hu, Xiong, Campos, & Overwijk, 2019). The representations of the candidate keyphrases are often captured by Convolution Neural Network (CNN) or Pre-trained Language Model. In the keyphrase importance estimation stage, many ranking algorithms are proposed to rank the candidate keyphrases (Jiang, Hu, & Li, 2009). This process is often reformulated as a span classification problem, where keyphrases are extracted by a binary classifier. Recently, Sun, Xiong, Liu, Liu, and Bao (2020) jointly employs a chunking network and a ranking network to balance the estimation of keyphrase quality and salience for keyphrase extraction. Song, Jing, and Xiao (2021) proposes a new keyphrase importance estimation approach from multiple perspectives, including syntactic accuracy, information saliency, and concept consistency simultaneously.

Despite their effectiveness, all of these extraction methods can only extract keyphrases that appear in the source document. They fail to extract absent keyphrases that do not appear in the source document. Researchers attempt to formalize the keyphrase extraction as a unified keyphrase generation problem using generic Sequence-to-Sequence model (Meng et al., 2021). Chowdhury, Rossiello, Glass, Mihindukulasooriya, and Gliozzo (2022) adopts a generative pre-trained language model BART to generate both present and absent keyphrases. Ye, Cai, Gui, and Zhang (2021) proposes a heterogeneous graph-based approach that can capture explicit knowledge from related references for absent keyphrase generation. In contrast to supervised methods which require large amount of labeled training corpora, unsupervised methods are more flexible and adaptable by extracting keyphrases based on information from the input documents themselves. In this paper, we focus on unsupervised keyphrase extraction methods.

### 6.3. Unsupervised keyphrase extraction

Researchers have developed a series of solutions for unsupervised keyphrase extraction (UKE). The most traditional methods are statistics-based models (Campos et al., 2018). These methods extract keyphrases by analyzing the word frequency feature, position feature or linguistic feature of an article. Topic-based models (Liu, Li, Zheng, & Sun, 2009) mine keyphrases according to the probability distribution of documents.

Graph-based models, which use centrality scoring algorithm to determine the importance of candidate keyphrases, are the most popular approaches. A document is represented as a graph where words or phrases are nodes of the graph and edges between them are weighted by their similarities. Specifically, the early work TextRank (Mihalcea, 2004) translates the keyphrase extraction into the ranking of nodes of the graph. After that, a variety of works are proposed to extend TextRank. SingleRank (Wan & Xiao, 2008)

adopts co-occurrences of words as edge weights. TopicRank (Bougouin et al., 2013) assigns a significance score to each topic by clustering candidate keyphrases. Boudin (2018) proposes MultipartiteRank, which encodes topical information within a multipartite graph structure. Recently, Vega-Oliveros, Gomes, Milios, and Berton (2019) analyzes nine centrality measures to find the optimal combination of word rankings for keyphrase extraction. Ushio, Liberatore, and Camacho-Collados (2021) conducts a quantitative analysis of statistical and graph-based term weighting scheme for keyphrase extraction. Their interesting findings could serve as a reference for future studies to understand the advantages and disadvantages of each method in different settings.

Embedding-based models (Papagiannopoulou & Tsoumakas, 2018; Wang, Jin, Zhu, & Goutte, 2016) map natural text into low-dimension vector representation space. Thanks to the development of representation learning, these methods have achieved potential performance. Specifically, EmbedRank (Bennani-Smires et al., 2018) extract keyphrases by measuring the similarities between embeddings of candidate keyphrases and the document. SIFRank (Sun, Qiu et al., 2020) extends EmbedRank by replacing the static embedding with a deep contextualized representations from the pre-trained language model. Zhang et al. (2022) address the issue of sequence length mismatch between candidate keyphrases and the document by computing the similarity between a masked document and source document. These existing models ignore the local information by calculating similarities between document and candidate keyphrases only. To this end, UKERank (Liang, Wu, Li, & Li, 2021a) calculates the similarity between candidate keyphrase and the entire document as global similarity, and combines it with boundary-aware centrality which is regarded as local salience to extract keyphrase. There also exists some works that resort to pre-trained language models to extract keyphrases. Ding and Luo (2021) investigates the accumulated self-attention and cross-attention from the pre-trained language model for unsupervised keyphrase extraction.

As we pointed out earlier, the salience of sentences has not been exploited by all of these existing methods, which limits their performance for this task. We hypothesize that salient sentences in a document are particularly important as they are most likely to contain keyphrases. To this end, we propose two model variants, Sentence-then-Keyphrase Ranking Model and Hierarchical Graph Representation Ranking Model, to characterize the salience of sentences and further improve the performance of unsupervised keyphrase extraction.

## 7. Conclusion

In this paper, we demonstrate that the existing works ignore the salience of sentences for unsupervised keyphrase extraction. Our work is among the earliest studies to exploit the salience of sentences for unsupervised keyphrase extraction by modeling hierarchical multi-granularity features. We propose two model variants, a Sentence-then-Keyphrase Ranking Model and a Hierarchical Graph Representation Ranking Model, to characterize the salience of sentence and further improve the performance of unsupervised keyphrase extraction. Extensive experiments are conducted to validate and explain the effectiveness of our model. Our study also shows positive synergy between the salient sentence extraction task and the keyphrase extraction task, suggesting that better integration of these two tasks is a promising direction. This will be a useful insight to practitioners of the field. Future work will further exploit the strong relations between the salient sentence extraction task and the keyphrase extraction task in a supervised manner.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

We thank the editors and anonymous referees for their efforts in reviewing this work.

## References

Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., & Jaggi, M. (2018). Simple unsupervised keyphrase extraction using sentence embeddings. In A. Korhonen, & I. Titov (Eds.), *Proceedings of the 22nd conference on computational natural language learning, CoNLL 2018, Brussels, Belgium, October 31–November 1, 2018* (pp. 221–229). Association for Computational Linguistics.

Boudin, F. (2018). Unsupervised keyphrase extraction with multipartite graphs. In M. A. Walker, H. Ji, & A. Stent (Eds.), *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: human language technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1–6, 2018, vol. 2* (pp. 667–672). Association for Computational Linguistics.

Bougouin, A., Boudin, F., & Daille, B. (2013). TopicRank: Graph-based topic ranking for keyphrase extraction. In *Sixth international joint conference on natural language processing, IJCNLP 2013, Nagoya, Japan, October 14–18, 2013* (pp. 543–551). Asian Federation of Natural Language Processing / ACL.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A. M., Nunes, C., & Jatowt, A. (2018). YAKE! collection-independent automatic keyword extractor. In G. Pasi, B. Piwowarski, L. Azzopardi, & A. Hanbury (Eds.), *Lecture notes in computer science: vol. 10772, Advances in information retrieval - 40th European conference on IR research, ECIR 2018, Grenoble, France, March 26–29, 2018, proceedings* (pp. 806–810). Springer.

Cheng, Y., Li, S., Liu, B., Zhao, R., Li, S., Lin, C., et al. (2021). Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing, vol. 1*.

Chowdhury, M. F. M., Rossiello, G., Glass, M. R., Mihindukulasooriya, N., & Gliozzo, A. (2022). Applying a generic sequence-to-sequence model for simple and effective keyphrase generation. CoRR arXiv:2201.05302.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, vol. 1* (pp. 4171–4186). Association for Computational Linguistics.

Ding, H., & Luo, X. (2021). AttentionRank: Unsupervised keyphrase extraction using self and cross attentions. In M. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021, virtual event/punta cana, Dominican Republic, 7–11 November, 2021* (pp. 1919–1928). Association for Computational Linguistics.

Dong, Y., Romascanu, A., & Cheung, J. C. K. (2021). Discourse-aware unsupervised summarization for long scientific documents. In P. Merlo, J. Tiedemann, & R. Tsarfaty (Eds.), *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume, EACL 2021, Online, April 19–23, 2021* (pp. 1089–1102). Association for Computational Linguistics.

Florescu, C., & Caragea, C. (2017a). A position-biased PageRank algorithm for keyphrase extraction. In S. Singh, & S. Markovitch (Eds.), *Proceedings of the thirty-first AAAI conference on artificial intelligence, February 4–9, 2017, San Francisco, California, USA* (pp. 4923–4924). AAAI Press.

Florescu, C., & Caragea, C. (2017b). PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In R. Barzilay, & M. Kan (Eds.), *Proceedings of the 55th annual meeting of the association for computational linguistics, ACL 2017, Vancouver, Canada, July 30–August 4, vol. 1* (pp. 1105–1115). Association for Computational Linguistics.

Gu, X., Wang, Z., Bi, Z., Meng, Y., Liu, L., Han, J., et al. (2021). UCPhrase: Unsupervised context-aware quality phrase tagging. In F. Zhu, B. C. Ooi, C. Miao (Eds.), *KDD '21: The 27th ACM SIGKDD conference on knowledge discovery and data mining, virtual event, Singapore, August 14–18, 2021* (pp. 478–486). ACM.

Hasan, K. S., & Ng, V. (2014). Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd annual meeting of the association for computational linguistics, ACL 2014, June 22–27, 2014, Baltimore, MD, USA, vol. 1* (pp. 1262–1273). The Association for Computer Linguistics.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the conference on empirical methods in natural language processing, EMNLP 2003, Sapporo, Japan, July 11–12, 2003*.

Jiang, X., Hu, Y., & Li, H. (2009). A ranking approach to keyphrase extraction. In J. Allan, J. A. Aslam, M. Sanderson, C. Zhai, & J. Zobel (Eds.), *Proceedings of the 32nd annual international ACM SIGIR conference on research and development in information retrieval, SIGIR 2009, Boston, MA, USA, July 19–23, 2009* (pp. 756–757). ACM.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of tricks for efficient text classification. In M. Lapata, P. Blunsom, & A. Koller (Eds.), *Proceedings of the 15th conference of the european chapter of the association for computational linguistics, EACL 2017, Valencia, Spain, April 3–7, 2017, vol. 2* (pp. 427–431). Association for Computational Linguistics.

Kim, S. N., Medelyan, O., Kan, M., & Baldwin, T. (2010). SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In K. Erk, & C. Strapparava (Eds.), *Proceedings of the 5th international workshop on semantic evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15–16, 2010* (pp. 21–26). The Association for Computer Linguistics.

Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *JMLR workshop and conference proceedings*: *vol. 32*, *Proceedings of the 31th international conference on machine learning, ICML 2014, Beijing, China, 21–26 June 2014* (pp. 1188–1196). JMLR.org.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., et al. (2020). BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In D. Jurafsky, J. Chai, N. Schluter, J. R. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics, ACL 2020, Online, July 5–10, 2020* (pp. 7871–7880). Association for Computational Linguistics.

Liang, X., Li, J., Wu, S., Li, M., & Li, Z. (2021). Improving unsupervised extractive summarization by jointly modeling facet and redundancy. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Liang, X., Wu, S., Li, M., & Li, Z. (2021a). Improving unsupervised extractive summarization with facet-aware modeling. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), *Findings of ACL*: *vol. ACL/IJCNLP 2021*, *Findings of the association for computational linguistics: ACL/IJCNLP 2021, online event, August 1–6, 2021* (pp. 1685–1697). Association for Computational Linguistics.

Liang, X., Wu, S., Li, M., & Li, Z. (2021b). Unsupervised keyphrase extraction by jointly modeling local and global context. In M. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021, virtual event/punta cana, Dominican Republic, 7–11 November, 2021* (pp. 155–164). Association for Computational Linguistics.

Liang, X., Wu, S., Li, M., & Li, Z. (2021c). Unsupervised keyphrase extraction by jointly modeling local and global context. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 155–164).

Liu, Z., Li, P., Zheng, Y., & Sun, M. (2009). Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 conference on empirical methods in natural language processing, EMNLP 2009, 6–7 August 2009, Singapore, a meeting of SIGDAT, a special interest group of the ACL* (pp. 257–266). ACL.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., et al. (2019). RoBERTa: A robustly optimized BERT pretraining approach. CoRR arXiv:1907.11692.

Meng, R., Yuan, X., Wang, T., Zhao, S., Trischler, A., & He, D. (2021). An empirical study on neural keyphrase generation. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tür, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, & Y. Zhou (Eds.), *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2021, Online, June 6–11, 2021* (pp. 4985–5007). Association for Computational Linguistics.

Mihalcea, R. (2004). Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the 42nd annual meeting of the association for computational linguistics, Barcelona, Spain, July 21–26*. ACL.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing , EMNLP 2004, a meeting of SIGDAT, a special interest group of the ACL, held in conjunction with ACL 2004, 25–26 July 2004, Barcelona, Spain* (pp. 404–411). ACL.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In Y. Bengio, & Y. LeCun (Eds.), *1st international conference on learning representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, workshop track proceedings*.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank citation ranking: Bringing order to the web.*: *Technical report*, Stanford InfoLab.

Pagliardini, M., Gupta, P., & Jaggi, M. (2018). Unsupervised learning of sentence embeddings using compositional n-gram features. In M. A. Walker, H. Ji, & A. Stent (Eds.), *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1–6, 2018, vol. 1* (pp. 528–540). Association for Computational Linguistics.

Papagiannopoulou, E., & Tsoumakas, G. (2018). Local word vectors guiding keyphrase extraction. *Information Processing and Management*, *54*(6), 888–902.

Peng, K., Yin, C., Rong, W., Lin, C., Zhou, D., & Xiong, Z. (2021). Named entity aware transfer learning for biomedical factoid question answering. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *19*(4), 2365–2376.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, a meeting of SIGDAT, a special interest group of the ACL* (pp. 1532–1543). ACL.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., et al. (2018). Deep contextualized word representations. In M. A. Walker, H. Ji, A. Stent (Eds.), *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1–6, 2018, vol. 1* (pp. 2227–2237). Association for Computational Linguistics.

Saxena, A., Mangal, M., & Jain, G. (2020). KeyGames: A game theoretic approach to automatic keyphrase extraction. In D. Scott, N. Bel, & C. Zong (Eds.), *Proceedings of the 28th international conference on computational linguistics, COLING 2020, Barcelona, Spain (Online), December 8–13, 2020* (pp. 2037–2048). International Committee on Computational Linguistics.

Song, S., Huang, H., & Ruan, T. (2019). Abstractive text summarization using LSTM-CNN based deep learning. *Multimedia Tools and Applications*, *78*(1), 857–875.

Song, M., Jing, L., & Xiao, L. (2021). Importance estimation from multiple perspectives for keyphrase extraction. In M. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021, virtual event/punta cana, Dominican Republic, 7–11 November, 2021* (pp. 2726–2736). Association for Computational Linguistics.

Sun, Y., Qiu, H., Zheng, Y., Wang, Z., & Zhang, C. (2020). SIFRank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model. *IEEE Access*, *8*, 10896–10906.

Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., et al. (2020). ERNIE 2.0: A continual pre-training framework for language understanding. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020* (pp. 8968–8975). AAAI Press.

Sun, S., Xiong, C., Liu, Z., Liu, Z., & Bao, J. (2020). Joint keyphrase chunking and salience ranking with BERT. CoRR arXiv:2004.13639.

Ushio, A., Liberatore, F., & Camacho-Collados, J. (2021). Back to the basics: A quantitative analysis of statistical and graph-based term weighting schemes for keyword extraction. In M. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021, virtual event/punta cana, Dominican Republic, 7–11 November, 2021* (pp. 8089–8103). Association for Computational Linguistics.

Vega-Oliveros, D. A., Gomes, P. S., Milios, E. E., & Berton, L. (2019). A multi-centrality index for graph-based keyword extraction. *Information Processing and Management*, *56*(6).

Wan, X., & Xiao, J. (2008). Single document keyphrase extraction using neighborhood knowledge. In D. Fox, & C. P. Gomes (Eds.), *Proceedings of the twenty-third AAAI conference on artificial intelligence, AAAI 2008, Chicago, Illinois, USA, July 13–17, 2008* (pp. 855–860). AAAI Press.

Wang, Y., Fan, Z., & Rosé, C. P. (2020). Incorporating multimodal information in open-domain web keyphrase extraction. In B. Webber, T. Cohn, Y. He, & Y. Liu (Eds.), *Proceedings of the 2020 conference on empirical methods in natural language processing, EMNLP 2020, Online, November 16–20, 2020* (pp. 1790–1800). Association for Computational Linguistics.

Wang, Y., Jin, Y., Zhu, X., & Goutte, C. (2016). Extracting discriminative keyphrases with learned semantic hierarchies. In N. Calzolari, Y. Matsumoto, & R. Prasad (Eds.), *COLING 2016, 26th international conference on computational linguistics, proceedings of the conference: technical papers, December 11–16, 2016, Osaka, Japan* (pp. 932–942). ACL.

Xiong, L., Hu, C., Xiong, C., Campos, D., & Overwijk, A. (2019). Open domain web keyphrase extraction beyond language modeling. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019* (pp. 5174–5183). Association for Computational Linguistics.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada* (pp. 5754–5764).

Ye, J., Cai, R., Gui, T., & Zhang, Q. (2021). Heterogeneous graph neural networks for keyphrase generation. In M. Moens, X. Huang, L. Specia, & S. W. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing, EMNLP 2021, virtual event/punta cana, Dominican Republic, 7–11 November, 2021* (pp. 2705–2715). Association for Computational Linguistics.

Zhang, L., Chen, Q., Wang, W., Deng, C., Zhang, S., Li, B., et al. (2022). MDERank: A masked document embedding rank approach for unsupervised keyphrase extraction. In S. Muresan, P. Nakov, A. Villavicencio (Eds.), *Findings of the association for computational linguistics: ACL 2022, Dublin, Ireland, May 22–27, 2022* (pp. 396–409). Association for Computational Linguistics.

Zhao, T., Yan, Z., Cao, Y., & Li, Z. (2021). A unified multi-task learning framework for joint extraction of entities and relations. In *Thirty-Fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, Virtual Event, February 2–9, 2021* (pp. 14524–14531). AAAI Press.

Zheng, H., & Lapata, M. (2019). Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 6236–6247). Florence, Italy: Association for Computational Linguistics.