# Mitigating Over-generation for Unsupervised Keyphrase Extraction with Heterogeneous Centrality Detection

**Mingyang Song, Pengyu Xu, Yi Feng, Huafeng Liu***, **Liping Jing***
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University, Beijing, China
mingyang.song@bjtu.edu.cn

## Abstract

Over-generation errors occur when a keyphrase extraction model correctly determines a candidate keyphrase as a keyphrase because it contains a word that frequently appears in the document but at the same time erroneously outputs other candidates as keyphrases because they contain the same word. To mitigate this issue, we propose a new heterogeneous centrality detection approach (CentralityRank), which extracts keyphrases by simultaneously identifying both implicit and explicit centrality within a heterogeneous graph as the importance score of each candidate. More specifically, CentralityRank detects centrality by taking full advantage of the content within the input document to construct graphs that encompass semantic nodes of varying granularity levels, not limited to just phrases. These additional nodes act as intermediaries between candidate keyphrases, enhancing inter-phrase relevance. Furthermore, we introduce a novel adaptive boundary-aware regularization that can leverage the position information of candidate keyphrases, thus influencing the importance of candidate keyphrases. Extensive experimental results demonstrate the superiority of CentralityRank over recent state-of-the-art unsupervised keyphrase extraction baselines on three benchmark datasets.

## 1 Introduction

Keyphrase Extraction (KE) is the task of extracting a set of salient and relevant phrases (e.g., "information extraction", "natural language processing", "ontology", "intelligent analysis", and "semantic analysis" in Table 1) from the source document, which is a fundamental task in natural language processing (Song et al., 2023b). Because of their succinct and accurate expression, keyphrase extraction is helpful for various applications (Song et al., 2021a; Liu et al., 2009; Kim et al., 2013; Song et al., 2022b, 2023a; Liu et al., 2023; Xiao et al., 2023, 2021; Lyu et al., 2023).

**Part of the Input Document**:
**Information** extraction using Natural Language Processing tools focuses on extracting explicitly stated **information** from textual material (...) to perform intelligent analysis on the **information**, we provide an ontology, which describes the domain of the extracted **information**, in addition to rules that govern the classification and interpretation of added elements. The ontology is at the core of an interactive system that assists analysts with the collection, extraction, organization, analysis and retrieval of **information**, with the topic of "terrorism financing" as a case study. User interaction provides valuable assistance in assigning meaning to extracted **information**. The system is designed as a set of tools to provide the user with the flexibility and power (...)

**Ground-Truth Keyphrases**:
**information** extraction; nlp; intelligent analysis; ontology; modeling; natural language processing; owl; semantic analysis

**One of the Correct Keyphrases**: **information** extraction
**Possible Over-Generation Cases**: retrieval of **information**, stated **information**, the extracted **information**

Table 1: The input document with its corresponding keyphrases. Underlined words indicate ground-truth keyphrases, while high-frequency words in the document are highlighted in bold red. Here, we introduce the over-generation error with an example. As shown in the above case, one of the ground-truth keyphrases is " **information** extraction". In this setting, if "stated **information**" is extracted, then it is an example of the over-generation error.

Extracting keyphrases with less redundancy is challenging due to repetitions often present in documents. These repetitions can lead to different types of redundant keyphrase extraction, which can be broadly categorized into three cases. The first and simplest case is complete repetition, which can be easily resolved through a straightforward deduplication process. The second case involves alias repetition, which can be effectively managed by introducing external knowledge into the process. The third case represents the most problematic scenario, known as over-generation errors, as discussed previously (Bahuleyan and El Asri, 2020). As mentioned before, over-generation errors occur when a model correctly predicts a candidate as a keyphrase because it contains a word that frequently appears in its corresponding document but at the same time erroneously outputs other candidates as keyphrases because they contain the same word.

*Corresponding Author

*Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16349–16359
December 6-10, 2023 ©2023 Association for Computational Linguistics

Recall that for many keyphrase extraction models, it is not easy to reject a non-keyphrase containing a word with a high term frequency, e.g., many models score a candidate keyphrase by summing the score of each associated word. Generally, to be more concrete, consider the source document in Table 1, where the ground-truth keyphrases are underlined. As we can see, the word "information" has a significant presence in the document. Many existing keyphrase extraction systems not only correctly predict "information extraction" as a keyphrase but also erroneously predict "stated information" as a keyphrase, yielding over-generation errors.

Unsupervised keyphrase extraction models primarily consist of two steps: candidate keyphrase generation and keyphrase importance estimation (Song et al., 2023d,c). Candidate keyphrase generation involves extracting a list of words or phrases that can potentially serve as keyphrases using specific heuristics (Wan and Xiao, 2008a; Song et al., 2021b, 2022a, 2023f). Keyphrase importance estimation, on the other hand, is responsible for estimating the importance scores of these candidates and determining which of them are indeed correct keyphrases. Recently, pre-trained language models such as ELMo, BERT, and RoBERTa (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019) have emerged as pivotal technologies, achieving remarkable advancements in various natural language tasks. Concretely, these models build upon the concept of word embeddings by learning contextual representations from extensive corpora adopting a language modeling objective. Leveraging these advancements, many contemporary unsupervised keyphrase extraction models (Liang et al., 2021; Song et al., 2022c) incorporate pre-trained language models as the embedding layer and calculate the semantic relatedness between candidates and the document to determine their importance scores. Despite the significant progress made by these methods, addressing the over-generation errors in the keyphrase extraction task is essential.

In this paper, we propose a heterogeneous centrality detection model for unsupervised keyphrase extraction (CentralityRank) to mitigate the issue of over-generation errors. Specifically, CentralityRank extracts keyphrases within the source document by detecting implicit and explicit centrality within a heterogeneous graph. Instead of constructing graphs solely based on phrase-level nodes for modeling explicit interactions between keyphrases, we introduce additional semantic units as nodes (e.g., word and document) in the graph to model the implicit interactions among candidate keyphrases. These additional nodes serve as intermediaries, enhancing the implicit relationships between candidate keyphrases. Essentially, each additional node represents a unique relationship between the candidate keyphrases it encompasses. During interactions within the heterogeneous graph, these additional nodes are considered alongside phrase nodes to estimate the importance of each candidate keyphrase. While more advanced semantic units like entities or topics can be utilized, this paper employs words, phrases, and documents as semantic units for simplicity. The advantages of constructing a heterogeneous graph include (a) enabling different candidate phrases to interact with each other explicitly, (b) utilizing contextual information at varying granularities to estimate the importance of each candidate keyphrase accurately, (c) incorporating additional types of semantic nodes easily, such as topics and entities, and (d) enhancing the importance of each candidate keyphrase by updating within a heterogeneous graph with graph-based models. Furthermore, to enhance the robustness of our model, a novel adaptive boundary-aware regularization is proposed to optimize the importance scores of candidate keyphrases via the position information. Extensive experiments demonstrate that CentralityRank consistently outperforms recent unsupervised keyphrase extraction baselines across benchmark datasets. The main contributions of this paper can be summarized as follows:

- We fully model contextual information within a heterogeneous graph, which captures relationships among candidate keyphrases. This graph comprises not only phrase nodes but also other semantic units. Although we use word, phrase, and document nodes in this paper, more superior semantic units (e.g., topics and entities) can be incorporated.

- CentralityRank is adaptable by adding different nodes, such as seamlessly transitioning from single-document keyphrase extraction to multi-document keyphrase extraction.

- CentralityRank consistently outperforms all existing baselines across three benchmark keyphrase extraction datasets. Ablation studies and qualitative analysis show the effectiveness of our proposed model.

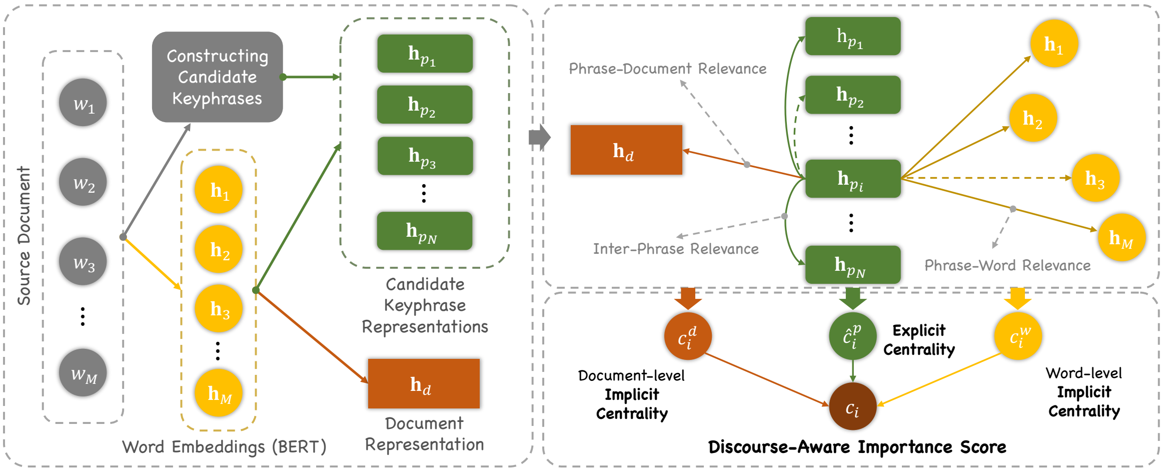

Figure 1: The overall architecture of CentralityRank model.

## 2 Methodology

### 2.1 Overview

The overall architecture of our model is depicted in Figure 1. Specifically, CentralityRank comprises two primary steps: candidate keyphrase generation and keyphrase importance estimation. In the first step, we leverage natural language linguistics to generate candidate keyphrases from the input document. The second step involves embedding candidate keyphrases and their corresponding document into a low-dimensional semantic space using a pre-trained language model BERT (Devlin et al., 2019) and then estimates and ranks the importance of each candidate keyphrase by detecting heterogeneous centrality, ultimately leading to the extraction of the top-ranked candidates as keyphrases. We present more details of these components in the following sections.

### 2.2 Candidate Keyphrase Generation

As the primary contribution of CentralityRank lies in proposing an importance estimation model, we have employed the same candidate keyphrase generation strategy as the unsupervised keyphrase extraction baselines (Liang et al., 2021; Song et al., 2023d) for a fair comparison. Consequently, we rely on Stanford CoreNLP Tools[1] for tasks such as tokenization, part-of-speech tagging, and noun phrase chunking. To generate candidates, we utilize a regular expression, $< NN.|JJ >< NN.* >$, designed to extract noun phrases as candidates through the Python package NLTK[2].

### 2.3 Heterogeneous Centrality Detection

Typically, a keyphrase extraction model is expected to produce a ranked list of candidate keyphrases, necessitating the ranking of generated candidate keyphrases based on their relevance score to the input document. Existing studies (Bennani-Smires et al., 2018; Sun et al., 2020; Ding and Luo, 2021; Song et al., 2023d; Liang et al., 2021; Song et al., 2023c) suggest that implicit interactions (e.g., utilizing contextual information of the input document as an intermediary) and explicit interactions (e.g., employing the pairwise ranking strategy) contribute to the importance estimation of each candidate keyphrase.

To concurrently model implicit and explicit interactions between all candidate keyphrases, we construct a heterogeneous graph based on the input document. This graph is formed by assessing the relevance between different levels of information granularity (including word, phrase, and document) concerning candidate keyphrases. This approach enables us to capture implicit and explicit centralities associated with each candidate keyphrase. Ultimately, we aggregate multiple centralities to derive the importance scores of candidate keyphrases for ranking and extracting keyphrases. Furthermore, we introduce a position encoding approach and propose a novel adaptive boundary-aware regularization to optimize the discourse-aware importance scores of candidate keyphrases.

#### 2.3.1 Text Representation

After obtaining all candidate keyphrases $\mathbf{P} = \{p_1, ..., p_n, ..., p_N\}$ of the input document $\mathcal{D}$, we adopt the pre-trained language model BERT (De-

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

[2] https://github.com/nltk

vlin et al., 2019) as the embedding layer to obtain word representations $\mathcal{H}$ for the input document $\mathcal{D}$ as follows,

$$\begin{aligned}\mathcal{H} &= [\mathbf{h}_1^\top, ..., \mathbf{h}_m^\top, ..., \mathbf{h}_M^\top]^\top \\ &= \text{BERT}(\{w_1, ..., w_m, ..., w_M\}),\end{aligned} \tag{1}$$

where $\mathbf{h}_m$ indicates the representation of the $m$-th word in the input document. Subsequently, we utilize word representations to derive candidate keyphrase representations. Given the nature of the keyphrase extraction task, it is typically desired that the extracted keyphrases can effectively encapsulate the central semantics of the input document (Song et al., 2023b). To achieve this, we obtain candidate keyphrase representations by utilizing the max pooling operation, which is a straightforward and efficient parameter-free approach. Then, the representation of the $i$-th candidate keyphrase $p_i$ can be calculated as follows,

$$\mathbf{h}_{p_i} = \text{Max-Pooling}(\{\mathbf{h}_1, ..., \mathbf{h}_k, ..., \mathbf{h}_{|p_i|}\}), \tag{2}$$

where $\mathbf{h}_{p_i}$ indicates the representation of the $i$-th candidate keyphrase and $|p_i|$ indicates the length of the $i$-th candidate keyphrase $p_i$. Specifically, $\mathbf{h}_k$ represents the word in the document associated with the candidate keyphrase $p_i$. At the same time, we use the same method to capture the central semantics of the document $\mathbf{h}_d$.

### 2.3.2 Explicit Centrality Detection

Calculating the inter-phrase relevance is a crucial aspect of keyphrase extraction and has been the focus of numerous existing approaches. One intuitive approach is representing these relations within a graph, offering a more intricate structure for capturing inter-phrase relationships. Moreover, the importance score of a candidate keyphrase is frequently determined by its degree of relevance to other candidate keyphrases within the document. To explicitly model these inter-phrase relationships, we begin by computing the relevance between the $i$-th and $j$-th candidates to initialize the edges within the graph,

$$e_{i,j} = \frac{\mathbf{h}_{p_i}\mathbf{h}_{p_j}^\top}{\sqrt{d}}. \tag{3}$$

Here, $d$ represents the dimension of $\mathbf{h}_{p_i}$, and $\sqrt{d}$ is a scalar. While other similarity measurement methods, such as cosine similarity, can be considered, empirical observations indicate that the straightforward dot-product tends to yield superior results. Then, the explicit centrality of the $i$-th candidate keyphrase can be computed as follows:

$$c_i^p = \sum_{j=1, j\neq i}^{N-1} (e_{i,j} - \delta_i), \tag{4}$$

$$\delta_i = \frac{1}{N-1} \sum_{j=1, j\neq i}^{N-1} e_{i,j}. \tag{5}$$

Here, $c_i^p$ represents the explicit centrality of the $i$-th candidate keyphrase, and $\delta_i$ denotes the average centrality of the $i$-th candidate keyphrase. In practice, we consider $\delta_i$ as a de-noising factor, effectively filtering out edges with low relevance between candidate keyphrases within the heterogeneous graph.

### 2.3.3 Implicit Centrality Detection

Intuitively, a candidate keyphrase that receives high scores for informativeness, indicating its ability to capture the central idea of the input document in which it appears, is more likely to be considered a keyphrase (Tomokiyo and Hurst, 2003; Song et al., 2023b). Consequently, we leverage words and their corresponding document nodes as intermediaries between candidate keyphrases, effectively enhancing the implicit inter-phrase relationships. To capture the inter-phrase relevance implicitly, we commence by computing the relevance between the $i$-th candidate and words to establish the initial edges within the graph,

$$e_{i,m} = \text{cosine}(\mathbf{h}_{p_i}, \mathbf{h}_m), \tag{6}$$

where $e_{i,m}$ represents the relevance between the $i$-th candidate keyphrase and the $m$-th word in the input document (establishing the edges within the graph). Next, we can calculate the implicit centrality of the $i$-th candidate phrase as follows:

$$c_i^w = \sum_{m=1}^{M} \max(0, e_{i,m} - \frac{1}{M}\sum_{m=1}^{M} e_{i,m}), \tag{7}$$

where $c_i$ indicates the implicit centrality of the $i$-th candidate keyphrase. Here, $\frac{1}{M}\sum_{m=1}^{M} e_{i,m}$ denotes the de-noise coefficient, which is used to avoid the deviation of centrality caused by meaningless words in the document.

To identify the most relevant keyphrases, we leverage the entire information in the input document. This information serves as an intermediary for computing the relevance of each candidate keyphrase, which is used to initialize the edges connecting candidate keyphrases to the whole input document:

$$e_i = \frac{1}{||\mathbf{h}_d - \mathbf{h}_{p_i}||_1}, \tag{8}$$

where $|| \cdot ||_1$ represents the Manhattan Distance. Given that each data pair in all existing keyphrase extraction datasets is centered around a single document and its associated keyphrases, we can directly utilize the phrase-document relevance as the document-level implicit centrality. Furthermore, our graph is highly adaptable and can naturally extend from single-document to multi-document scenarios by adding document nodes. Ultimately, we aggregate the implicit centrality as follows:

$$c_i^d = c_i^w \odot e_i, \tag{9}$$

where $c_i^d$ denotes the final implicitly centrality of the $i$-th candidate keyphrase.

#### 2.3.4 Phrase Position Encoding

In various specialized domain text documents, such as scientific and news articles, keyphrases often tend to appear prominently at the beginning or front of the document (Florescu and Caragea, 2017a,b; Liang et al., 2021). Therefore, we incorporate positional information as a regularization mechanism to penalize centralities of candidate keyphrases. And it can be calculated as follows,

$$\rho_i = \frac{e^{\frac{1}{i}}}{\sum_{i=1}^{N} e^{\frac{1}{i}}}, \tag{10}$$

where $\rho_i$ is the position regularization of the $i$-th candidate keyphrase. By employing the aforementioned regularization technique, we can elevate the importance scores of candidate keyphrases that are positioned at the beginning of the document.

#### 2.3.5 Adaptive Boundary-Aware Regularization

Traditional centrality computation assumes that the contribution of the importance score of each candidate keyphrase in the input document is not influenced by their relative position, and the similarities between the two graph nodes are symmetrical. However, human intuition suggests that phrases located at the beginning or end of a document should carry greater importance than others. Therefore, in this paper, we introduce an adaptive boundary-aware regularization to enhance explicit centrality. Accordingly, Equation 4 can be reformulated as follows,

$$\begin{aligned}\hat{c}_i^p = &\sum_{\mathit{ABAR}(i)<\mathit{ABAR}(j)} \left(\frac{\mathbf{h}_{p_i}\mathbf{h}_{p_j}^\top}{\sqrt{d}} - \delta_i\right) + \\ &\sum_{\mathit{ABAR}(i)\geq \mathit{ABAR}(j)} \lambda\left(\frac{\mathbf{h}_{p_i}\mathbf{h}_{p_j}^\top}{\sqrt{d}} - \delta_i\right).\end{aligned} \tag{11}$$

where $\lambda$ is a weighting factor. Here, $\mathit{ABAR}(\cdot)$ indicates the proposed adaptive boundary-aware function, which can be formulated as

$$\mathit{ABAR}(i) = \min(i, \frac{\sqrt{N}}{\log N}(N - i)). \tag{12}$$

Here, $N$ indicates the length of the input document.

#### 2.3.6 Discourse-Aware Importance Score

Upon acquiring the explicit and implicit centrality scores for each candidate keyphrase, we consolidate them into a single importance score as follows,

$$c_i = c_i^d \cdot \hat{c}_i^p \cdot \rho_i \tag{13}$$

where $c_i$ indicates the discourse-aware importance score of the $i$-th candidate keyphrase. In the end, we rank all candidate keyphrases with their importance scores and select top-ranked K candidates as keyphrases of the input document.

## 3 Experiments

We conduct experiments to demonstrate the effectiveness of our proposed CentralityRank model. In this section, we introduce our experimental settings, including datasets, evaluation metrics, baselines, and implementation details.

### 3.1 Datasets

In this paper, we carry out experiments on three benchmark keyphrase extraction datasets, which includes **DUC2001** (Wan and Xiao, 2008b), **Inspec** (Hulth, 2003), and **SemEval2010** (Kim et al., 2010). The DUC2001 dataset (Wan and Xiao, 2008b) is a collection of 308 long length news articles with average 828.4 tokens. The Inspec dataset (Hulth, 2003) contains 2,000 short scientific abstracts. Specifically, similar to the previous work (Sun et al., 2020; Liang et al., 2021), we use 500 test documents and the version of uncontrolled annotated keyphrases as the ground-truth label. The SemEval2010 dataset (Kim et al., 2010) contains ACM full length papers. Consistent with previous

| **Model** | **DUC2001** | | | **Inspec** | | | **SemEval2010** | | |
|---|---|---|---|---|---|---|---|---|---|
| | F1@5 | F1@10 | F1@15 | F1@5 | F1@10 | F1@15 | F1@5 | F1@10 | F1@15 |
| **Statistical Models** | | | | | | | | | |
| TF-IDF (Jones, 2004) | 9.21 | 10.63 | 11.06 | 11.28 | 13.88 | 13.83 | 2.81 | 3.48 | 3.91 |
| YAKE (Campos et al., 2018) | 12.27 | 14.37 | 14.76 | 18.08 | 19.62 | 20.11 | 11.76 | 14.4 | 15.19 |
| **Graph-based Models** | | | | | | | | | |
| TextRank (Mihalcea and Tarau, 2004) | 11.80 | 18.28 | 20.22 | 27.04 | 25.08 | 36.65 | 3.80 | 5.38 | 7.65 |
| SingleRank (Wan and Xiao, 2008b) | 20.43 | 25.59 | 25.70 | 27.79 | 34.46 | 36.05 | 5.90 | 9.02 | 10.58 |
| TopicRank (Bougouin et al., 2013) | 21.56 | 23.12 | 20.87 | 25.38 | 28.46 | 29.49 | 12.12 | 12.90 | 13.54 |
| PositionRank (Florescu and Caragea, 2017b) | 23.35 | 28.57 | 28.60 | 28.12 | 32.87 | 33.32 | 9.84 | 13.34 | 14.33 |
| MultipartiteRank (Boudin, 2018) | 23.20 | 25.00 | 25.24 | 25.96 | 29.57 | 30.85 | 12.13 | 13.79 | 14.92 |
| **Embedding-based Models** | | | | | | | | | |
| EmbedRank (Bennani-Smires et al., 2018) | 24.02 | 28.12 | 28.82 | 31.51 | 37.94 | 37.96 | 3.02 | 5.08 | 7.23 |
| KeyGames (Saxena et al., 2020) | 24.42 | 28.28 | 29.77 | 32.12 | 40.48 | 40.94 | 11.93 | 14.35 | 14.62 |
| SIFRank (Sun et al., 2020) | 30.88 | 33.37 | 32.24 | 28.49 | 36.77 | 38.82 | - | - | - |
| JointGL (Liang et al., 2021) | 28.62 | 35.52 | 36.29 | 32.61 | 40.17 | 41.09 | 13.02 | 19.35 | 21.72 |
| MDERank (Zhang et al., 2022) | 23.31 | 26.65 | 26.42 | 27.85 | 34.36 | 36.40 | 13.05 | 18.27 | 20.35 |
| HGUKE (Song et al., 2023d) | 31.31 | 37.24 | 38.31 | **34.18** | **41.05** | **42.16** | 14.07 | 20.52 | 23.10 |
| **CentralityRank** | **31.63** | **37.77** | **38.77** | 32.99 | 40.93 | 41.73 | **15.51** | **21.39** | **23.83** |

Table 2: Performance on DUC2001, Inspec and SemEval2010 test sets. The best results are in bold.

studies (Song et al., 2023d; Liang et al., 2021), we leverage the 100 test documents and the combined set of author-and reader-annotated keyphrases.

### 3.2 Evaluation Metrics

To evaluate the quality of extracted keyphrases, specifically their relevance to the source document, we compare the extracted set of keyphrases with the keyphrases in their corresponding ground-truth data. Following the previous studies (Liang et al., 2021; Song et al., 2023d,c; Ding and Luo, 2021; Bennani-Smires et al., 2018; Saxena et al., 2020; Sun et al., 2020), we evaluate the performance of our model using the F1-measure at the top-K keyphrases (F1@K) and apply stemming (using the Porter Stemmer[3]) to both the extracted keyphrases and the ground truth. More specifically, we present the F1@5, F1@10, and F1@15 evaluation scores for our model and baselines across three benchmark keyphrase extraction datasets.

### 3.3 Baselines

We compare our model with the recent state-of-the-art unsupervised keyphrase extraction models, which contain statistics-based, graph-based, and embedding-based models. The statistics-based ranking models include TF-IDF (Jones, 2004) and YAKE (Campos et al., 2018). The graph-based ranking models include TextRank (Mihalcea and Tarau, 2004), SingleRank (Wan and Xiao, 2008b), TopicRank (Bougouin et al., 2013), PositionRank (Florescu and Caragea, 2017b), and MultipartiteRank (Boudin, 2018). The embedding-based unsupervised keyphrase extraction models:

- **EmbedRank** (Bennani-Smires et al., 2018) ranks candidate keyphrases by directly measuring the semantic similarity between candidate keyphrases and the input document.
- **KeyGames** (Saxena et al., 2020) introduces game theoretic into unsupervised keyphrase extraction to address the over-generation issue and extract better keyphrases.
- **SIFRank** (Sun et al., 2020) improves the traditional embeddings (e.g., word2vec) from EmbedRank with a pre-trained language model as the external knowledge.
- **MDERank** (Zhang et al., 2022) designs a document-level candidate keyphrase representations and optimize the whole model by a self-supervised learning framework.
- **JointGL** (Liang et al., 2021) proposes a BERT-based unsupervised keyphrase extraction approach which jointly models global and local context information to estimate the importance scores of candidate keyphrases.
- **HGUKE** (Song et al., 2023d) proposes to model the phrase-document relevance via the highlight of the document instead of the entire document.

In addition, the results of the selected baselines are reported in their corresponding papers.

[3] https://tartarus.org/martin/PorterStemmer/

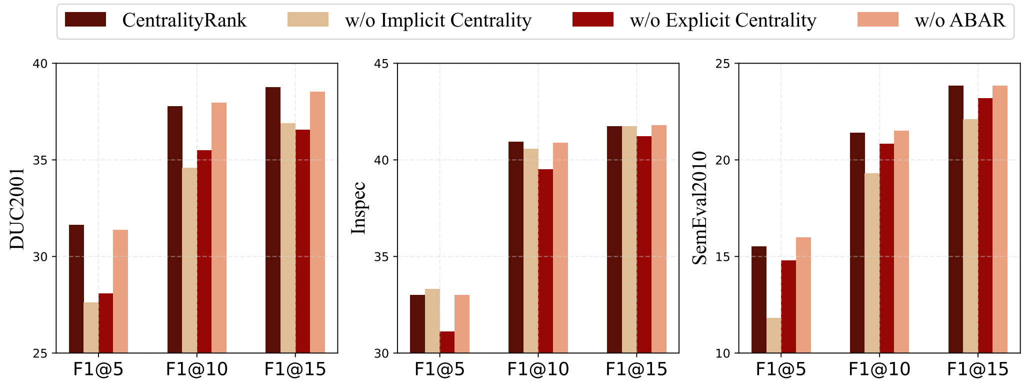

Figure 2: The results of ablation experiments on three benchmark datasets. For example, CentralityRank w/o ABAR indicates the model without the adaptive boundary-aware regularization.

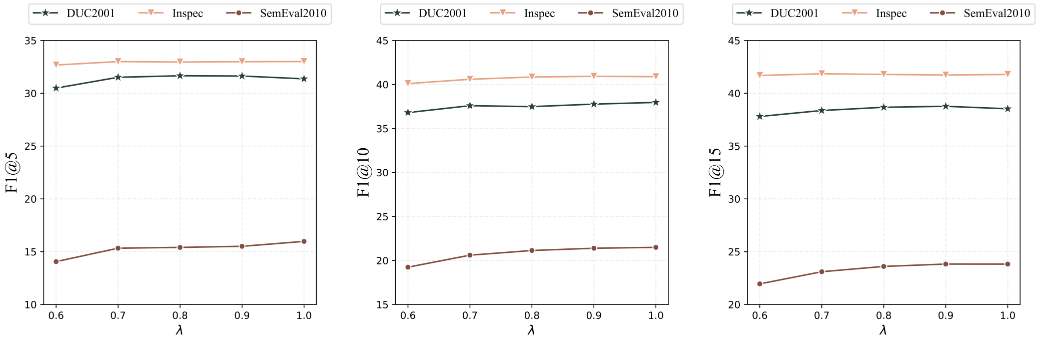

Figure 3: Performance with different $\lambda$ of our model CentralityRank on three benchmark datasets.

### 3.4 Implementation Details

We adopt the pre-trained language model BERT[4] (Devlin et al., 2019) as the backbone of our model, initialized from their pre-trained weights. The maximum document length is 512 due to BERT limitations, and documents are truncated to this size. The dimension of BERT-based representation is set to 768. For all datasets, we use the same hyperparameter, i.e., $\lambda = 0.9$.

## 4 Results and Analysis

### 4.1 Overall Performance

Table 2 presents the results obtained from baseline models and our CentralityRank model across three benchmark datasets (DUC2001, Inspec, and SemEval2010). In summary, the results highlight the significant superiority of CentralityRank over recent state-of-the-art unsupervised keyphrase extraction baselines. More specifically, our model demonstrates substantial improvements on all evaluation metrics compared to traditional unsupervised keyphrase extraction models, including statistical-based and graph-based approaches. This outcome is unsurprising as embedding-based methods leverage pre-trained language models as their foundation, which yields superior representations for accurately estimating the importance scores of candidate keyphrases. Furthermore, our model performs significantly better on all evaluation metrics than the embedding-based baseline models, especially for the long document dataset (SemEval2010). This emphasizes the effectiveness of detecting implicit and explicit centrality in estimating the importance of candidate keyphrases for fully modeling contextual information of the input document to enhance the performance.

### 4.2 Ablation Test

In this section, several ablation experiments are conducted to analyze the effect of different components in CentralityRank. The ablation experiments on three datasets are shown in Figure 2. In total, four variants of the proposed model (CentralityRank, CentralityRank w/o Implicit Centrality Detection, CentralityRank w/o Explicit Centrality Detection, and CentralityRank w/o ABAR) were involved in the ablation experiment. Concretely, Cen-

[4] https://huggingface.co/transformers/index.html

**Part of the Input Document**:
Some of this year's ***drought*** in the ***Midwest*** may have been caused by ***ocean temperature abnormalities*** near the equator in the Pacific Ocean, according to a ***new computer study*** reported Thursday. Such droughts could be anticipated if the ***temperature abnormalities*** turn out to be predictable, one of the authors said in the report appearing in Friday's issue of Science magazine. The authors are Kevin E. Trenberth and Grant W. Branstator of the National Center for ***Atmospheric Research*** in Boulder, ... They noted that when asked what caused the drought that hit much of North America in 1988, meteorologists often reply " the jet stream was displaced northward of its usual position so that storms, which tend to track along the path of the jet stream, were similarly displaced northward." " Such an answer is, however, just a brief description of the ***weather patterns*** associated with the drought but does not get at the cause. ... ... In this period, ***Pacific Ocean temperatures*** ranged up to 5.4 degrees ... ...

**Target Keyphrase**:
(1) ***drought***; (2) ***Midwest***; (3) ***ocean temperature abnormalities***; (4) ***new computer study***; (5) ***temperature abnormalities***; (6) ***Atmospheric Research***; (7) ***weather patterns***; (8) ***Pacific Ocean temperatures***;

**CentralityRank**:
Top 1-5: (1) ***ocean temperature abnormalities***; (2) ***weather patterns***; (3) jet stream; (4) temperature pattern; (5) ***temperature abnormalities***;
Top 6-10: (6) low pressure centers; (7) Pacific Ocean; (8) low pressure systems; (9) ***Pacific Ocean temperatures***; (10) below-normal equatorial temperature;
Top 11-15: (11) ***drought***; (12) surface temperatures; (13) North America; (14) global atmosphere; (15) ***new computer study***;

Table 3: Example of keyphrase extraction results on the Inspec (contains part of the input document and target keyphrases). Phrases in orange and bold are keyphrases predicted by our models. Top5, Top10, and Top15 extracted keyphrases are provided by our models without repetitions.

tralityRank w/o ABAR indicates the model without the adaptive boundary-aware regularization.

Based on the results illustrated in Figure 2, it becomes evident that both the implicit and explicit centrality detection modules contribute significantly to enhancing model performance in most cases. This observation reinforces the importance of assessing the importance scores of candidate keyphrases from multiple perspectives. Moreover, the results also illustrate that employing intermediaries to influence the relationship modeling among candidate keyphrases within a graph, consequently affecting the estimation of importance, represents an effective strategy.

### 4.3 Sensitivity of Hyper-Parameter $\lambda$

In this section, we analyze the hyper-parameter $\lambda$ and present the results for each dataset in Figure 3. Our proposed model, CentralityRank, delivers optimal performance when $\lambda$ is set to 0.9. More specifically, we attribute this trend to the adaptive boundary-aware regularization, which elevates the significance of candidate keyphrases that initially appear in the document while diminishing the importance of identical keyphrases that surface later, indirectly mitigating over-generation errors.

### 4.4 Case Study

In this section, we randomly sample a document from the DUC2001 dataset to examine the extracted keyphrases generated by our CentralityRank model. As seen in the results presented in Table 3, the keyphrases extracted by CentralityRank not only encompass commonly occurring words within the document but also include those that appear only once. This observation shows the effectiveness of our importance estimation approach, achieved through detecting both implicit and explicit centrality within a heterogeneous graph of the document. This further helps mitigate the issue of over-generating keyphrases during extraction.

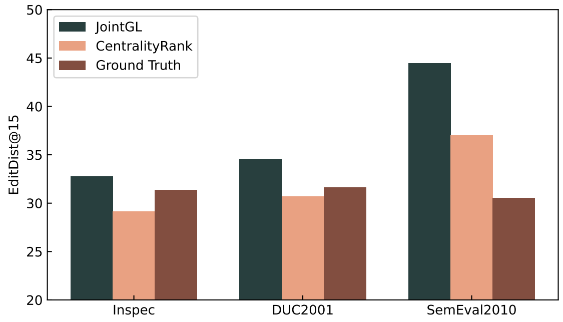

Figure 4: Redundancy of extraction results. Concretely, EditDist@15 is used to measure the redundancy among the top 15 extracted keyphrases via the EditDist metric.

### 4.5 Redundancy Evaluation

To evaluate redundancy, we introduce an evaluation metric inspired by previous work (Bahuleyan and El Asri, 2020; Song et al., 2023e,c) called **EditDist**. We employ the *fuzzywuzzy* library[5], which provides a score ranging from 0 to 100, where a score of 100 signifies an exact match between keyphrases. Using this metric, we compute the pairwise Levenshtein Distance between the extracted keyphrases.

Figure 4 and Figure 5 illustrate the redundancy in the extraction results between our model and the baseline model JointGL. The results show that our approach significantly reduces the redundancy of the extracted keyphrases, thereby achieving higher-quality keyphrases.

[5] https://github.com/seatgeek/fuzzywuzzy

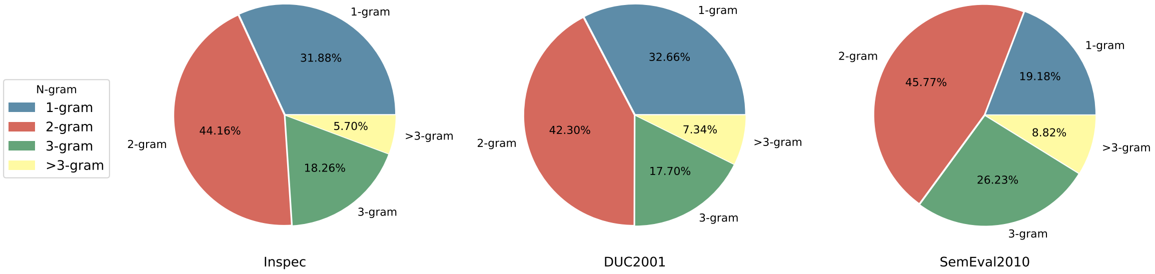

Figure 5: The ratio of redundancy contained in extracted keyphrases at different lengths. Specifically, the Edit-Dist@15 evaluation metric calculates the total redundancy.

Furthermore, it is notable that our model CentralityRank substantially reduces redundancy in the keyphrase extraction results of the long document dataset, i.e., the SemEval2010 dataset.

## 5 Related Work

Most of the existing keyphrase extraction models can be broadly categorized into four main groups: statistics-based, topic-based, graph-based, and embedding-based models. Statistics-based models (Salton and Buckley, 1988; Witten et al., 1999) generally identify keyphrases by estimating the importance of candidate keyphrases through various statistical features. These features may include word frequency, phrase position, linguistic characteristics of natural language, and more. Topic-based models (Liu et al., 2009, 2010) typically leverage topic information to determine whether a candidate keyphrase qualifies as a keyphrase. They often involve methods that consider the topical relevance of keyphrases to the document. Graph-based models (Mihalcea and Tarau, 2004; Grineva et al., 2009) represent the input document as a graph, where nodes correspond to words or phrases, and edges indicate relationships between them. Candidate keyphrases are ranked based on graph-based measures such as centrality or similarity. Embedding-based models (Bennani-Smires et al., 2018; Song et al., 2023d) make use of pre-trained word embeddings to capture semantic representations of keyphrases. Then, they calculate the importance of each candidate keyphrase by its representation. These categories encompass a wide range of techniques and approaches for keyphrase extraction, each with its strengths and limitations. Researchers in the field often choose a specific type or combine elements from multiple categories to develop effective keyphrase extraction models.

Unlike the existing models, to address the over-generation error, our model selects keyphrases with less redundancy by detecting implicit and explicit centrality within a heterogeneous graph to filter noises in the input document.

## 6 Conclusion

In this paper, we propose a heterogeneous centrality detection model, which incorporates implicit and explicit centrality to identify and select keyphrases in the source document. Additionally, we leverage the position information of candidate keyphrases to optimize the discourse-aware importance score of each candidate keyphrase. Our experiments yield compelling results that highlight the superiority of our model when compared to existing SOTA keyphrase extraction baselines. We validate our model on three benchmark datasets, demonstrating its robust performance and effectiveness in improving keyphrase extraction quality.

## 7 Limitations

There are still some limitations to our work. In the future, we plan to investigate enhancing the quality of keyphrase representations and improving the accuracy of importance estimation to enhance the performance of keyphrase extraction. One possible way is to leverage task-specific (i.e., the keyphrase extraction task) language models (Kulkarni et al., 2022) to obtain better representations.

## 8 Acknowledgments

This work was partly supported by the Fundamental Research Funds for the Central Universities (2019JBZ110); the National Natural Science Foundation of China under Grant 62176020; the National Key Research and Development Program (2020AAA0106800); the Beijing Natural Science Foundation under Grant L211016; CAAI-Huawei MindSpore Open Fund; and Chinese Academy of Sciences (OEIP-O-202004).

## References

Hareesh Bahuleyan and Layla El Asri. 2020. Diverse keyphrase generation with neural unlikelihood training. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5271–5287. International Committee on Computational Linguistics.

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *CoNLL*, pages 221–229. Association for Computational Linguistics.

Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite graphs. In *NAACL-HLT (2)*, pages 667–672. Association for Computational Linguistics.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. In *IJCNLP*, pages 543–551. Asian Federation of Natural Language Processing / ACL.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. Yake! collection-independent automatic keyword extractor. In *ECIR*, volume 10772 of *Lecture Notes in Computer Science*, pages 806–810. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

Haoran Ding and Xiao Luo. 2021. Attentionrank: Unsupervised keyphrase extraction using self and cross attentions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1928.

Corina Florescu and Cornelia Caragea. 2017a. A position-biased pagerank algorithm for keyphrase extraction. In *AAAI*, pages 4923–4924. AAAI Press.

Corina Florescu and Cornelia Caragea. 2017b. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *ACL (1)*, pages 1105–1115. Association for Computational Linguistics.

Maria P. Grineva, Maxim N. Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In *WWW*, pages 661–670. ACM.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *EMNLP*.

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60(5):493–502.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *SemEval@ACL*, pages 21–26. The Association for Computer Linguistics.

Youngsam Kim, Munhyong Kim, Andrew Cattle, Julia Otmakhova, Suzi Park, and Hyopil Shin. 2013. Applying graph-based keyword extraction to document retrieval. In *IJCNLP*, pages 864–868. Asian Federation of Natural Language Processing / ACL.

Mayank Kulkarni, Debanjan Mahata, Ravneet Arora, and Rajarshi Bhowmik. 2022. Learning rich representation of keyphrases from text. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 891–906, Seattle, United States. Association for Computational Linguistics.

Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Unsupervised keyphrase extraction by jointly modeling local and global context. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 155–164, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *HLT-NAACL*, pages 620–628. The Association for Computational Linguistics.

Xin Liu, Shijing Wang, Kairui Zhou, Yilin Lyu, Mingyang Song, Liping Jing, Tieyong Zeng, and Jian Yu. 2023. Vmf loss: Exploring a scattered intra-class hypersphere for few-shot learning. In *Machine Learning and Knowledge Discovery in Databases: Research Track: European Conference, ECML PKDD 2023*, page 454–470, Berlin, Heidelberg. Springer-Verlag.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *CoRR*, abs/1907.11692.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *EMNLP*, pages 366–376. ACL.

Yilin Lyu, Xin Liu, Mingyang Song, Xinyue Wang, Yaxin Peng, Tieyong Zeng, and Liping Jing. 2023. Recognizable information bottleneck. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023*, pages 4028–4036. ijcai.org.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *EMNLP*, pages 404–411. ACL.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*, pages 2227–2237. Association for Computational Linguistics.

Gerard Salton and Chris Buckley. 1988. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523. Also available in Sparck Jones and Willett (1997).

Arnav Saxena, Mudit Mangal, and Goonjan Jain. 2020. Keygames: A game theoretic approach to automatic keyphrase extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2037–2048.

Mingyang Song, Yi Feng, and Liping Jing. 2022a. Hyperbolic relevance matching for neural keyphrase extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022*, pages 5710–5720. Association for Computational Linguistics.

Mingyang Song, Yi Feng, and Liping Jing. 2022b. A preliminary exploration of extractive multi-document summarization in hyperbolic space. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4505–4509. ACM.

Mingyang Song, Yi Feng, and Liping Jing. 2022c. Utilizing BERT intermediate layers for unsupervised keyphrase extraction. In *5th International Conference on Natural Language and Speech Processing, ICNLSP 2022*, pages 277–281. Association for Computational Linguistics.

Mingyang Song, Yi Feng, and Liping Jing. 2023a. Hisum: Hyperbolic interaction model for extractive multi-document summarization. In *Proceedings of the ACM Web Conference 2023, WWW 2023*, pages 1427–1436. ACM.

Mingyang Song, Yi Feng, and Liping Jing. 2023b. A survey on recent advances in keyphrase extraction from pre-trained language models. In *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 2108–2119. Association for Computational Linguistics.

Mingyang Song, Haiyun Jiang, Lemao Liu, Shuming Shi, and Liping Jing. 2023c. Unsupervised keyphrase extraction by learning neural keyphrase set function. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2482–2494. Association for Computational Linguistics.

Mingyang Song, Liping Jing, Yi Feng, Zhiwei Sun, and Lin Xiao. 2021a. Hybrid summarization with semantic weighting reward and latent structure detector. In *Proceedings of The 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, pages 1739–1754. PMLR.

Mingyang Song, Liping Jing, and Lin Xiao. 2021b. Importance Estimation from Multiple Perspectives for Keyphrase Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2726–2736. Association for Computational Linguistics.

Mingyang Song, Huafeng Liu, Yi Feng, and Liping Jing. 2023d. Improving embedding-based unsupervised keyphrase extraction by incorporating structural information. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1041–1048. Association for Computational Linguistics.

Mingyang Song, Huafeng Liu, and Liping Jing. 2023e. Improving diversity in unsupervised keyphrase extraction with determinantal point process. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 4294–4299. Association for Computing Machinery.

Mingyang Song, Lin Xiao, and Liping Jing. 2023f. Learning to extract from multiple perspectives for neural keyphrase extraction. *Comput. Speech Lang.*, 81:101502.

Yi Sun, Hangping Qiu, Yu Zheng, Zhongwei Wang, and Chaoran Zhang. 2020. Sifrank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model. *IEEE Access*, 8:10896–10906.

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. pages 33–40. Association for Computational Linguistics.

Xiaojun Wan and Jianguo Xiao. 2008a. Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *COLING*, pages 969–976.

Xiaojun Wan and Jianguo Xiao. 2008b. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, pages 855–860. AAAI Press.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Kea: Practical automatic keyphrase extraction. In *ACM DL*, pages 254–255. ACM.

Lin Xiao, Pengyu Xu, Mingyang Song, Huafeng Liu, Liping Jing, and Xiangliang Zhang. 2023. Triple alliance prototype orthotist network for long-tailed multi-label text classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2616–2628.

Lin Xiao, Xiangliang Zhang, Liping Jing, Chi Huang, and Mingyang Song. 2021. Does head label help for long-tailed multi-label text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14103–14111.

Linhan Zhang, Qian Chen, Wen Wang, Chong Deng, ShiLiang Zhang, Bing Li, Wei Wang, and Xin Cao. 2022. MDERank: A masked document embedding rank approach for unsupervised keyphrase extraction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 396–409, Dublin, Ireland. Association for Computational Linguistics.