HyperRank: Hyperbolic Ranking Model for Unsupervised Keyphrase Extraction

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

Abstract

Given the exponential growth in the number of documents on the web in recent years, there is an increasing demand for accurate models to extract keyphrases from such documents. Keyphrase extraction is the task of automatically identifying representative keyphrases from the source document. Typically, candidate keyphrases exhibit latent hierarchical structures embedded with intricate syntactic and semantic information. Moreover, the relationships between candidate keyphrases and the document also form hierarchical structures. Therefore, it is essential to consider these latent hierarchical structures when extracting keyphrases. However, many recent unsupervised keyphrase extraction models overlook this aspect, resulting in incorrect keyphrase extraction. In this paper, we address this issue by proposing a new hyperbolic ranking model (HyperRank). HyperRank is designed to jointly model global and local context information for estimating the importance of each candidate keyphrase within the hyperbolic space, enabling accurate keyphrase extraction. Experimental results demonstrate that HyperRank significantly outperforms recent state-of-the-art baselines.

1 Introduction

Keyphrase extraction is a natural language processing task that involves extracting a set of representative keyphrases from the source document (Hasan and Ng, 2014; Song et al., 2023b). This task is valuable for generating a concise summary or snippet of a web page. The core objective of keyphrase extraction is to condense information from the input document and preserve only the most essential details in the output. Distinguishing what information is important is a challenge in the keyphrase extraction task, primarily due to the fact that candidate keyphrases often exhibit latent hierarchical structures (Zhu et al., 2020; Song et al., 2022a), as

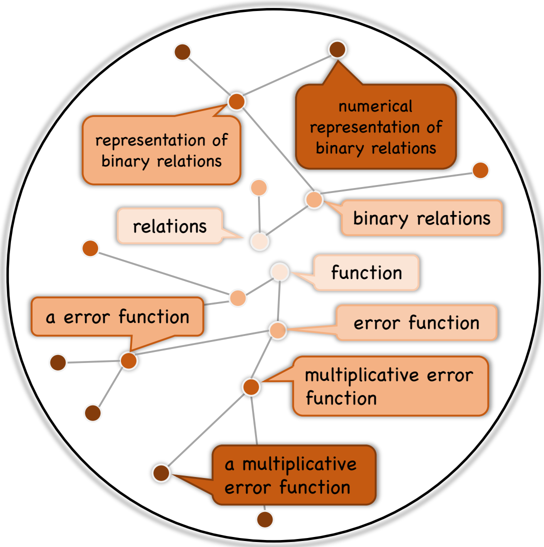

Figure 1: Illustration of the latent hierarchy hidden in candidate keyphrases by an example sentence "Numerical representation of binary relations with a multiplicative error function". Specifically, we present the latent hierarchical structures hidden among candidate keyphrases.

illustrated in Figure 1. Consequently, when extracting keyphrases, it is necessary to consider these latent hierarchical structures to accurately estimate the importance scores of candidate keyphrases.

Typically, most existing unsupervised keyphrase extraction models that rely on pre-trained word embeddings, as discussed in (Bennani-Smires et al., 2018; Sun et al., 2020; Saxena et al., 2020; Liang et al., 2021; Song et al., 2023d), can be categorized into two primary steps: candidate keyphrase generation and keyphrase importance estimation (Hasan and Ng, 2014; Song et al., 2021, 2023f,b). Specifically, the former step involves extracting a list of words or phrases from the source document that serve as candidate keyphrases through heuristic methods (Wan and Xiao, 2008a; Nguyen and Phan, 2009; Grineva et al., 2009; Song et al., 2023c; Liang et al., 2021). The latter step mainly

*Corresponding Author

Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 16070–16080
December 6-10, 2023 ©2023 Association for Computational Linguistics

consists of two components: text representation and importance calculation. Text representation is obtained using pre-trained language models such as BERT (Devlin et al., 2019). Subsequently, the importance scores of candidate keyphrases are estimated based on the pre-trained embeddings. This is achieved by calculating textual similarities between candidate keyphrases and the document using various distance measures, such as Manhattan distance, Euclidean distance, and Cosine distance. These measures are employed to determine which candidate keyphrases are the real keyphrases (Bennani-Smires et al., 2018; Sun et al., 2020; Song et al., 2023e; Liang et al., 2021; Song et al., 2023d).

In recent years, a significant breakthrough in natural language processing has been the widespread adoption of heavily pre-trained transformers designed for natural language modeling, exemplified by BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). These Pre-trained Language Models (PLMs) have proven to be formidable tools for various downstream tasks in natural language processing and information retrieval. As a result, they have evolved into indispensable components, serving as embedding layers to obtain contextualized representations in most NLP downstream tasks. Leveraging the advancements in text representation, embedding-based unsupervised keyphrase extraction models (Bennani-Smires et al., 2018; Sun et al., 2020; Liang et al., 2021; Zhang et al., 2022; Song et al., 2023c) have demonstrated promising results and have now established themselves as the new state-of-the-art benchmarks.

Candidate keyphrases frequently present an inherent hierarchical structure infused with intricate syntax and semantics (Zhu et al., 2020; Dai et al., 2020; Song et al., 2022a). In addition to candidate keyphrases, the relations between keyphrases and the document can also give rise to hierarchical structures. However, capturing such hierarchies, even in infinite dimensions within Euclidean space, poses a considerable challenge (Ganea et al., 2018a; Tifrea et al., 2019). Fortunately, hyperbolic space naturally lends itself to modeling these hierarchical structures, including tree-like characteristics (Zhu et al., 2020). This property makes hyperbolic space a valuable tool applied in various downstream natural language processing tasks (Chen et al., 2020; Song et al., 2023a; Chen et al., 2021).

Motivated by the phenomena mentioned above, in this paper, we explore the task of embedding-based unsupervised keyphrase extraction in the hyperbolic space. More specifically, we propose a new hyperbolic ranking model (HyperRank) designed to model the importance estimation in the hyperbolic space. In HyperRank, we first leverage the pre-trained language model BERT as the backbone of our model to enhance the quality of text representations, projecting these representations into the hyperbolic space. To estimate the importance of each candidate keyphrase accurately, we simultaneously consider two perspectives: phrase-document relevance and cross-phrase relevance. Concretely, the former calculates the relevance scores between candidate keyphrases and their corresponding document, while the latter calculates the relevance scores between all candidate keyphrases. We then combine these two relevance scores to determine the final importance scores of candidate keyphrases to rank and extract keyphrases. Our experimental results demonstrate that the proposed model HyperRank outperforms recent strong baselines significantly. We conduct various analyses on existing datasets to explore the characteristics of our model. Additionally, we discover that merely mapping representations to the hyperbolic space may likely mitigate the anisotropic issues in text representation in the keyphrase extraction task.

2 Related Work

We briefly introduce the recent progress on unsupervised keyphrase extraction and hyperbolic representation learning in this section.

2.1 Unsupervised Keyphrase Extraction

Unsupervised keyphrase extraction models can be categorized into statistics-based (Salton and Buckley, 1988; Witten et al., 1999), graph-based (Mihalcea and Tarau, 2004; Grineva et al., 2009), and embedding-based methods (Bennani-Smires et al., 2018; Sun et al., 2020; Saxena et al., 2020; Liang et al., 2021; Ding and Luo, 2021). To delve into more detail, embedding-based models have witnessed substantial advancements, mainly owing to the progress in representation learning. For instance, EmbedRank (Bennani-Smires et al., 2018) assesses candidate phrases by gauging their similarity through phrase and document embeddings. Subsequently, Sun et al. (2020) enhances the static embeddings from EmbedRank using a pre-trained language model. Liang et al. (2021) leverage local and global contextual information in Euclidean space to accurately estimate the importance score of each candidate keyphrase, consequently facilitating high-quality keyphrase extraction.

Diverging from existing models, we investigate unsupervised keyphrase extraction within the hyperbolic space rather than the conventional Euclidean space. More specifically, our model entails the initial mapping candidate keyphrases and their associated document representations into the same hyperbolic space. Subsequently, we employ the Poincaré Distance to calculate the phrase-document and cross-phrase relevance scores, aggregating and obtaining the importance score of each candidate keyphrase.

2.2 Hyperbolic Representation Learning

Recent advancements in representation learning, as illustrated by Nickel and Kiela (2017), have emphasized the superiority of hyperbolic space over Euclidean space, particularly in terms of representation capacity, especially when working with lower dimensions. Meanwhile, Ganea et al. (2018b) introduced a formalism for generalized operations in neural networks within the Poincaré ball, utilizing the framework of Möbius gyrovector space.

Later, recent research has showcased the advantages of hyperbolic space in various natural language processing tasks, including word embedding (Tifrea et al., 2019), machine translation (Ganea et al., 2018a), text classification (Chen et al., 2020), and text summarization (Song et al., 2022b, 2023a). Furthermore, previous research (Song et al., 2022a) has confirmed the effectiveness of modeling supervised keyphrase extraction models in hyperbolic space, yielding favorable results. In this paper, we further validate the feasibility of hyperbolic deep learning under an unsupervised setting.

3 Preliminary

Generally, hyperbolic space can be characterized using Riemannian geometry, as described in Hopper and Andrews (2011). In line with previous studies (Nickel and Kiela, 2017; Ganea et al., 2018b; Tifrea et al., 2019), we employ the Poincaré ball model and introduce an additional hyper-parameter denoted as c to modify the curvature of the Poincaré ball. This curvature modification is defined as $\mathbb{D}_c^n = \mathbf{x} \in \mathbb{R}^n : c\|\mathbf{x}\|^2 < 1, c \geq 0$. Concretely, the corresponding conformal factor is now represented as $\lambda_{\mathbf{x}}^c := \frac{2}{1-c\|\mathbf{x}\|^2}$. In practice, the choice of the parameter c allows us to balance between hyperbolic and Euclidean geometries. Notably, when c tends towards zero, all the formulas discussed below revert to their typical Euclidean forms. We provide a restatement of the fundamental mathematical operations for the generalized Poincaré ball model. For more comprehensive information, readers are encouraged to refer to Ganea et al. (2018b). Additionally, we present the closed-form expressions for several Möbius operations.

Möbius Addition. For a pair $\mathbf{x}, \mathbf{y} \in \mathbb{D}_c^n$, the Möbius addition is defined as,

$$\mathbf{x} \oplus_c \mathbf{y} = \frac{(1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c\|\mathbf{y}\|^2)\mathbf{x} + (1 - c\|\mathbf{x}\|^2)\mathbf{y}}{1 + 2c\langle \mathbf{x}, \mathbf{y} \rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}. \quad (1)$$

Exponential and Logarithmic Maps. To perform operations within the hyperbolic space, it is necessary to establish a mapping function from $\mathbb{R}^n$ to $\mathbb{D}_c^n$ in order to transfer Euclidean vectors into the hyperbolic space. We denote the tangent space of $\mathbb{D}_c^n$ at $\mathbf{x}$ as $T_{\mathbf{x}}\mathbb{D}_c^n$. The exponential map $\exp_{\mathbf{x}}^c(\cdot) : T_{\mathbf{x}}\mathbb{D}_c^n \to \mathbb{D}_c^n$ for $\mathbf{v} \neq 0$ is defined as follows:

$$\exp_{\mathbf{x}}^c(\mathbf{v}) = \mathbf{x} \oplus_c (\tanh(\sqrt{c}\frac{\lambda_{\mathbf{x}}^c\|\mathbf{v}\|}{2})\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}). \quad (2)$$

As the inverse of $\exp_{\mathbf{x}}^c(\cdot)$, the logarithmic map $\log_{\mathbf{x}}^c(\cdot) : \mathbb{D}_c^n \to T_x\mathbb{D}_c^n$ for $\mathbf{y} \neq \mathbf{x}$ can be calculated as follows:

$$\log_{\mathbf{x}}^c(\mathbf{y}) = \frac{2}{\sqrt{c}\lambda_{\mathbf{x}}^c}\tanh^{-1}(\sqrt{c}\|-\mathbf{x}\oplus_c\mathbf{y}\|)\frac{-\mathbf{x} \oplus_c \mathbf{y}}{\| -\mathbf{x} \oplus_c \mathbf{y}\|} \quad (3)$$

Poincaré Distance. The induced distance function is defined as,

$$d_c(\mathbf{x}, \mathbf{y}) = \frac{2}{\sqrt{c}}\text{arctanh}(\sqrt{c}\| - \mathbf{x} \oplus_c \mathbf{y}\|). \quad (4)$$

Here, $d_c(\cdot)$ denotes the Poincaré Distance. Note that with $c = 1$ one recovers the geodesic distance, while with $c \to 0$ we obtain the Euclidean distance, represented as, $\lim_{c\to 0}d_c(\mathbf{x}, \mathbf{y}) = 2\|\mathbf{x} - \mathbf{y}\|$.

4 Methodology

The overall architecture of our model HyperRank is illustrated in Figure 2. HyperRank comprises two main parts: candidate keyphrase generation and keyphrase importance estimation. In the first part, we employ natural language linguistic techniques to generate candidate keyphrases from the input document, as illustrated in Figure 3. The second part involves embedding candidate keyphrases and their corresponding document into a low-dimensional semantic space using the pre-trained

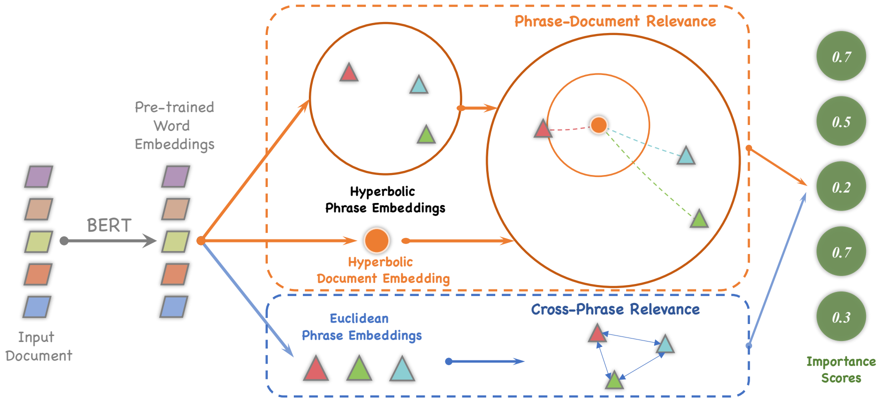

Figure 2: The overall architecture of our model HyperRank.

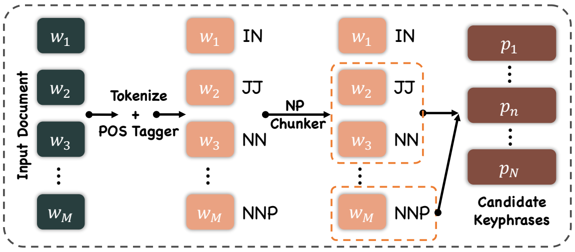

Figure 3: The procedure of our candidate keyphrase generation. (1) Get tokenized document and POS tags. (2) Extract noun phrases that consist of zero or more adjectives followed by one or multiple nouns.

language model BERT (Devlin et al., 2019). Then, HyperRank estimates and ranks the importance of each candidate keyphrase by modeling the context information of the input document from global and local perspectives, ultimately resulting in the extraction of the top-ranked candidates as keyphrases. Further details are shown in the following sections.

4.1 Candidate Keyphrase Generation

In this subsection, we describe the process of generating candidate keyphrases in our model HyperRank. In line with previous studies (Liang et al., 2021; Song et al., 2022c, 2023d), we employ Stanford CoreNLP Tools[1] for tasks such as tokenization, part-of-speech tagging, and noun phrase chunking. To generate candidate keyphrases, we use the regular expression $< NN.|JJ >< NN.* >$ and utilize the Python package NLTK[2] to extract noun phrases as candidate keyphrases. The detailed illustration of our candidate keyphrase generation process is in Figure 3.

[1] https://stanfordnlp.github.io/CoreNLP/
[2] https://github.com/nltk

4.2 Keyphrase Importance Estimation

In order to obtain the importance scores of candidate keyphrases, we first obtain the representations of the document and its corresponding candidate keyphrases through the pre-trained language model BERT. Subsequently, we derive the importance of candidate keyphrases through two distinct perspectives: phrase-document relevance and cross-phrase relevance. These two relevance are then combined to form the final importance scores for candidate keyphases. Finally, the importance scores are used to rank and extract keyphrases.

4.2.1 Text Representation

We adopt the pre-trained language model BERT (Devlin et al., 2019) to obtain word embeddings for the document $\mathcal{D} = \{w_1, ..., w_m, ..., w_M\}$,

$$\begin{aligned}\mathcal{H} &= [\mathbf{h}_1^\top, ..., \mathbf{h}_m^\top, ..., \mathbf{h}_M^\top]^\top \\ &= \mathrm{BERT}(\{w_1, ..., w_m, ..., w_M\}),\end{aligned} \tag{5}$$

where $\mathbf{h}_m$ indicates the representation of the m-th word in the source document. Next, we adopt word embeddings to obtain representations of candidate keyphrases. Considering the primary objective of the keyphrase extraction task, it is generally anticipated that the extracted keyphrases should effectively encapsulate the central semantics of the document (Hasan and Ng, 2014). To capture these central semantics within the candidate keyphrases, we obtain representations of the candidate keyphrases through max-pooling operation, a straightforward and parameter-free approach. Then, the representation of the i-th candidate keyphrase can be calculated as follows,

$$\mathbf{h}_{p_i} = \text{Max-Pooling}(\{\mathbf{h}_k, ..., \mathbf{h}_{k+|p_i|-1}\}), \tag{6}$$

where $\mathbf{h}_{p_i}$ is the i-th candidate keyphrase representation in the Euclidean space and $|p_i|$ indicates the length of p_i. Specifically, $\mathbf{h}_k$ represents the first word in the document associated with the candidate keyphrase p_i. Meanwhile, we adopt the average pooling operation to obtain the representation $\mathbf{h}_d$ of the source document $\mathcal{D}$.

4.2.2 Phrase-Document Relevance

As previously mentioned, candidate keyphrases often exhibit inherent hierarchical structures with complex syntax and semantics (Zhu et al., 2020; Dai et al., 2020; Nickel and Kiela, 2017). In addition to candidate keyphrases, latent hierarchical structures are hidden in the relations between candidate keyphrases and their corresponding document. However, capturing such hierarchical structures, even with infinite dimensions, proves challenging within the Euclidean space (Linial et al., 1995; Zhu et al., 2020; Chen et al., 2021).

In light of the aforementioned challenges, in this paper, we delve into unsupervised keyphrase extraction within the hyperbolic space. To elaborate, our model begins by mapping candidate keyphrases and their corresponding document representations into the same hyperbolic space. Subsequently, we compute the point-wise importance of the i-th candidate keyphrase by evaluating phrase-document relevance through the Poincaré distance, as illustrated below,

$$r_i^d = \frac{1}{d_c(\exp_{\mathbf{0}}^c(\mathbf{h}_d), \hat{\mathbf{h}}_{p_i})}. \tag{7}$$

Here, r_i^d indicates the phrase-document relevance of the i-th candidate keyphrase, $\exp_{\mathbf{0}}^c(\cdot)$ maps the Euclidean space inside the Poincaré ball, and $d_c(\cdot)$ denotes the Poincaré distance. Specifically, $\hat{\mathbf{h}}_{p_i}$ means the hyperbolic representation of the i-th candidate keyphrase and is computed as follows,

$$\begin{aligned} \hat{\mathbf{h}}_{p_i} &= \text{Max-Pooling}(\{\mathbf{h}_k, ..., \mathbf{h}_{k+|p_i|-1}\}). \\ &= \exp_{\mathbf{0}}^c(\{\mathbf{h}_k, ..., \mathbf{h}_{k+|p_i|-1}\}). \end{aligned} \tag{8}$$

The latent hierarchical relationships are implicitly modeled through the aforementioned process when determining the relevance between the document and its candidate keyphrases, effectively estimating the importance of candidate keyphrases.

4.2.3 Cross-Phrase Relevance

Typically, the phrase-document relevance is computed individually between the document and each candidate keyphrase, making it unable to distinguish which candidate keyphrases are superior to others (Liang et al., 2021; Song et al., 2023d). To identify more prominent keyphrases from all candidates, we introduce cross-phrase relevance, which assists in selecting the most relevant keyphrases from the candidate keyphrases and is calculated between the i-th candidate and all other candidates, as follows,

$$r_i^p = \sum_{j=1, j\neq i}^{N-1} \left(\frac{\mathbf{h}_{p_i}\mathbf{h}_{p_j}^\top}{\sqrt{d}} - \delta_i\right), \tag{9}$$

where $\frac{\mathbf{h}_{p_i}\mathbf{h}_{p_j}^\top}{\sqrt{d}}$ denotes the scaled semantic relatedness between the i-th candidate and the j-th candidate and d is the dimension of $\mathbf{h}_{p_i}$. δ_i means the average of semantic relatedness between the i-th candidate keyphrase with others ($\sum_{j=1,j\neq i}^{N-1} \frac{\mathbf{h}_{p_i}\mathbf{h}_{p_j}^\top}{\sqrt{d}}$), and we treat it as a de-noisy factor, which removes small semantic relatedness between all candidates.

4.2.4 Relevance Aggregation

After obtaining the two relevance scores, we combine them into a unified score, which serves as the importance score of each candidate keyphrase,

$$s_i = r_i^d \cdot r_i^p \tag{10}$$

where s_i indicates the importance score of the i-th candidate keyphrase.

4.2.5 Position Regularization

In various domain-specific text documents, such as scientific and news articles, keyphrases often have a tendency to appear at the beginning or front of the source document (Florescu and Caragea, 2017a,b; Liang et al., 2021; Song et al., 2023d). Therefore, we introduce positional information as a regularization mechanism to optimize the importance scores of candidate keyphrases. And the importance score of the i-th candidate keyphrase can be re-calculated as follows,

$$\rho_i = \frac{e^{\frac{1}{i}}}{\sum_{i=1}^{N} e^{\frac{1}{i}}}, \tag{11}$$

$$\hat{s}_i = \rho_i \cdot s_i. \tag{12}$$

where ρ_i is the position regularization of the i-th candidate keyphrase and $\hat{s}_i$ is the final importance score of the i-th candidate keyphrase. By implementing the position regularization strategy, we can enhance the importance scores of candidate

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									
EmbedRankd2v (Bennani-Smires et al., 2018)	24.02	28.12	28.82	31.51	37.94	37.96	3.02	5.08	7.23
EmbedRanks2v (Bennani-Smires et al., 2018)	27.16	31.85	31.52	29.88	37.09	38.40	5.40	8.91	10.06
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)	24.27	27.43	27.86	29.11	38.80	39.59	-	-	-
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
HyperRank	**32.68**	**39.18**	**40.21**	**33.35**	**40.79**	**42.12**	**14.79**	**21.33**	**24.20**
Improvement Gain (%)	14.18	10.30	10.80	2.27	1.54	2.51	13.59	10.23	11.42

Table 1: Performance on DUC2001, Inspec and SemEval2010 test sets. The best results are bolded in the table. The improvement gains (%) between **HyperRank** and the best baseline (JointGL) are listed. All the results in the table are obtained by using the last layer of BERT as the embedding layer.

keyphrases located at the start of the source document. Finally, we rank all candidates with their importance scores and extract top-ranked K candidates as keyphrases of the source document.

5 Experiments and Results

In this section, we conduct experiments to demonstrate the effectiveness of our proposed model HyperRank. Concretely, we first introduce our experimental settings and then present the experimental results and analysis.

5.1 Experimental Settings

Consistent with the previous studies (Liang et al., 2021; Song et al., 2023d), we conduct experiments on three benchmark keyphrase extraction datasets, namely, DUC2001 (Wan and Xiao, 2008b), Inspec (Hulth, 2003), and SemEval2010 (Kim et al., 2010). DUC2001 (Wan and Xiao, 2008b) consists of 308 news articles, each with an average of 828.4 tokens. Inspec (Hulth, 2003) contains 2,000 scientific abstracts. We use 500 test documents for our experiments and rely on the version of human-annotated keyphrases as the ground-truth label, aligning with the approach taken in prior research (Liang et al., 2021; Song et al., 2023d). SemEval2010 (Kim et al., 2010) is composed of ACM full-length papers. We evaluate our model on 100 test documents and employ the combined set of author- and reader-annotated keyphrases as the ground truth.

We follow the standard practice and evaluate the performance of our models using the F1-measure at the top-K keyphrases (F1@K). Additionally, we apply stemming to both the extracted keyphrases and the gold truth. Our reported metrics include F1@5, F1@10, and F1@15 for our models and baselines on three benchmark datasets.

To ensure a fair comparison, we employ the pre-trained language model BERT (bert-base-uncased) as the foundational element of our model, initializing it with its pre-trained weights. As BERT has a maximum document length constraint of 512 tokens, we truncate documents to adhere to this limit. In addition, the dimension of BERT-based representations is set to 768.

5.2 Results and Analysis

Table 1 presents the primary comparative results between our proposed models and recent state-of-the-art unsupervised keyphrase extraction baselines on three benchmark datasets, with higher scores indicating superior performance. Specifically, the results in Table 1 unequivocally show that HyperRank outperforms most state-of-the-art baselines by a margin across all evaluation metrics, indicating the effectiveness of our model.

Notably, HyperRank excels over statistical and graph-based models, primarily attributed to its utilization of the pre-trained language model BERT as the backbone, which results in enhanced representations and improved accuracy when calculating the importance of candidate keyphrases. These

HyperRank w/ PDR	R@5	R@10	R@15	P@5	P@10	P@15	F1@5	F1@10	F1@15
				DUC2001					
Manhattan Distance	14.68	21.50	27.66	22.54	17.00	14.91	17.56	18.75	19.11
Euclidean Distance	11.03	16.99	21.35	16.82	13.33	11.55	13.16	14.76	14.78
Cosine Distance	8.53	14.00	18.20	12.54	10.73	9.67	10.03	12.02	12.46
Poincaré Distance	**17.68**	**30.27**	**40.33**	**27.28**	**23.81**	**21.64**	**21.17**	**26.32**	**27.79**
				Inspec					
Manhattan Distance	19.55	32.26	43.31	31.80	27.29	25.75	22.68	27.82	30.66
Euclidean Distance	17.93	29.95	41.04	29.48	25.41	24.29	20.91	25.86	28.94
Cosine Distance	13.66	26.42	38.58	22.76	21.85	22.49	16.08	22.48	26.95
Poincaré Distance	**23.88**	**39.71**	**49.76**	**39.32**	**34.21**	**30.21**	**27.99**	**34.59**	**35.63**
				SemEval2010					
Euclidean Distance	5.43	9.35	12.39	12.80	11.10	10.00	7.58	10.08	10.99
Cosine Distance	3.64	5.35	7.56	8.80	6.50	6.13	5.10	5.81	6.70
Manhattan Distance	6.29	9.87	12.94	15.00	11.90	10.40	8.81	10.70	11.45
Poincaré Distance	**8.01**	**13.03**	**17.73**	**19.20**	**15.80**	**14.47**	**11.22**	**14.14**	**15.78**

Table 2: Performance on DUC2001, Inspec and SemEval2010 test sets. The best results are bolded in the table. Specifically, all the results in the table are obtained by using the last intermediate layer of BERT.

Model	SemEval2010		
	F1@5	F1@10	F1@15
HyperRank	**14.79**	**21.33**	**24.20**
HyperRank w/ CPR	11.85	18.61	22.03
HyperRank w/ PDR	11.22	14.14	15.78

Table 3: Ablation tests on the SemEval2010 test set. Specifically, HyperRank w/ PDR indicates that HyperRank only uses the phrase-document relevance.

findings also provide valuable insights into modeling text representation and importance estimation within the hyperbolic space.

5.3 Ablation Test

In this section, we evaluate the performance of each component of our model HyperRank. Therefore, we conduct several ablation experiments to study the impact of these components, the results of which are detailed in Table 3 on three benchmark datasets. Specifically, "HyperRank w/ CPR" represents our model without the calculation of phrase-document relevance. Overall, the results highlight the superiority of the individual components within our model in terms of overall performance, affirming the effectiveness of modeling text representations in the hyperbolic space and calculating the importance of candidate keyphrases via the Poincaré distance.

Observing the results from Table 3 and Table 1 jointly, compared to the baseline model (JointGL), "HyperRank w/ PDR" achieves comparable performance on three benchmark datasets in most cases. Notably, our findings indicate that phrase-document relevance holds greater significance than cross-phrase relevance in our models.

5.4 Impact of Different Distance Measures

In this paper, our model utilizes the Poincaré distance to estimate textual semantic similarity between candidate keyphrases and the entire document within the hyperbolic space. We also explored the application of various distance measures, including the Manhattan distance, Euclidean distance, and Cosine distance. The results associated with these different distance measures are detailed in Table 2, highlighting the advantage of the Poincaré distance. Meanwhile, we consider this advantage may partially come from the enhanced representation capacity of the hyperbolic space.

Furthermore, we observe that Cosine distance yields subpar results and could be unsuitable for calculating phrase-document relevance in our model. As mentioned, capturing such latent hierarchical structures remains challenging even within infinite dimensions in the Euclidean space.

5.5 Impact of Different Intermediate Layers

Previous studies (Kim et al., 2020; Rogers et al., 2020; Song et al., 2022a) have highlighted that the intermediate layers of the pre-trained language model BERT capture a diverse hierarchy of linguistic information, encompassing surface features in its lower layers, syntactic features in the middle layers, and semantic features in the higher layers. On the other hand, embedding-based unsupervised keyphrase extraction models rely on pre-trained language models to acquire embeddings and utilize these embeddings to calculate textual similarities as the importance scores for candidate keyphrases during keyphrase extraction. However, many existing embedding-based models focus solely on

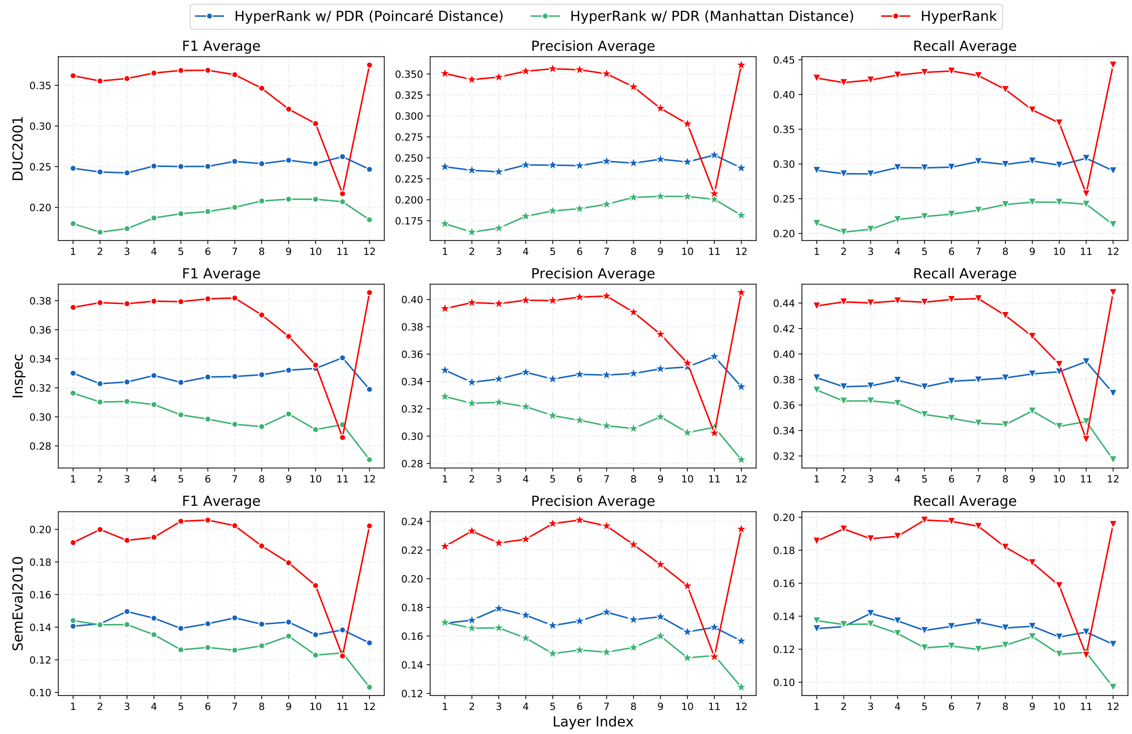

Figure 4: Results of using different intermediate layers of BERT on three benchmark datasets.

Document Embedding	**Phrase Embedding**	**DUC2001**		
		F1@5	F1@10	F1@15
CLS Token	Max-Pooling	32.02	37.11	38.83
	Average-Pooling	31.19	36.51	37.42
Average-Pooling	**Max-Pooling**	**32.68**	**39.18**	**40.21**
	Average-Pooling	32.23	37.34	38.96
Max-Pooling	Max-Pooling	29.98	36.65	37.06
	Average-Pooling	28.85	36.35	37.96

Table 4: Impact of different pooling operations for the document and its corresponding candidate keyphrases.

the last intermediate layer of the pre-trained language model to derive text representations, often overlooking the latent knowledge residing in the intermediate layers of BERT.

Therefore, in this paper, we evaluate our model HyperRank by employing different intermediate layers of BERT as the embedding layer across three benchmark datasets to verify the performance of HyperRank. The results, as depicted in Figure 4, reveal that the optimal performance may not necessarily be achieved by exclusively utilizing the last intermediate layer of BERT as the embedding layer. We attribute this to the varying language characteristics of keyphrases in different datasets, leading to the observed bias. This highlights the importance of embedding-based models considering different intermediate layers of BERT as potential embedding layers to tap into the full linguistic knowledge embedded in pre-trained language models. Intriguingly, we observed a notable performance drop when our model employed the 11th intermediate layer as the embedding layer. The reasons behind this phenomenon remain an interesting direction for future investigation.

Earlier studies (Jawahar et al., 2019) have noted that the pre-trained language model BERT predominantly captures phrase-level information in its lower layers and gradually diminishes this influence in higher layers. To delve deeper into the influence of employing different intermediate layers, we tested "HyperRank w/ PDR" using various distance measures. Interestingly, we find that "HyperRank w/ PDR" also yielded promising results when utilizing the earlier intermediate layers.

5.6 Impact of Different Pooling Operations

In our model, we empirically employ the average-pooling operation to derive the document repre-

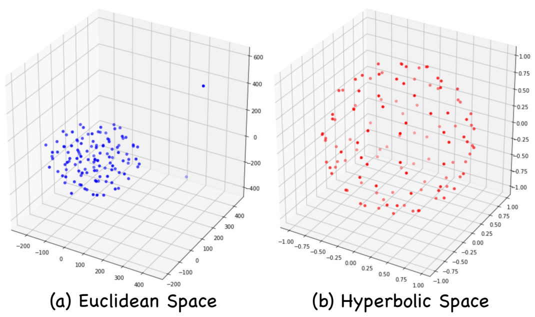

Figure 5: Visualization of candidate keyphrases embedded in the Euclidean and hyperbolic spaces.

sentation and utilize the max-pooling operation to derive the representations of candidate keyphrases. In Table 4, we present a comparison of various encoding methods, including max-pooling, average-pooling, and the "[CLS]" token. It is evident from the results that average-pooling is the most effective choice for document representation, while the max-pooling operation proves to be the most suitable choice for the representations of candidate keyphrases in our model.

5.7 Visualization of Representations in Different Semantic Spaces

A keyphrase extraction system correctly predicts a candidate keyphrase 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 have the same word. This phenomenon can be attributed to word frequency biases and further introduces the anisotropic[3] problem in learning representation for the embedding-based unsupervised keyphrase extraction models. To verify this opinion, we visualize the representations of candidate keyphrases in Euclidean and hyperbolic spaces, as illustrated in Figure 5.

The 3-dimensional representations of candidate keyphrases in Euclidean and hyperbolic spaces are depicted in Figure 7 (a) and (b), respectively. In general, it is evident that keyphrase representations in the Euclidean space exhibit relatively high density (anisotropy) while being widely dispersed in the hyperbolic space. Consequently, utilizing representations learned in the hyperbolic space may lead to more accurate estimations of importance scores between candidate keyphrases and their respective documents based on textual semantic similarities.

[3]"Anisotropic" means the pre-trained embeddings occupy a narrow cone in the vector space (Gao et al., 2019). This phenomenon is also observed in the pre-trained transformers like BERT.

6 Conclusion

In this paper, we propose a novel hyperbolic ranking model for unsupervised keyphrase extraction (HyperRank). HyperRank is designed to concurrently model text representation and importance estimation in the hyperbolic space. To ensure the accuracy of the importance estimation of each candidate keyphrase, HyperRank simultaneously utilizes global and local contextual information to estimate the relevance scores between candidate keyphrases and their corresponding documents. Afterward, HyperRank combines these two aspects of relevance to derive the final importance score for each candidate keyphrase, which is then used for ranking and keyphrase extraction. Our extensive experiments conclusively demonstrate that the proposed model HyperRank surpasses the performance of existing state-of-the-art unsupervised keyphrase extraction baselines.

7 Limitations

In this paper, we employ the pre-trained language model BERT as our encoder. However, it comes with a length limitation, typically allowing only 512 tokens for the length of the input document. Some of the keyphrase extraction datasets we work with have document lengths far exceeding this limit. Consequently, when dealing with long documents, there is a risk of information loss during encoding, leading to a skewed estimation of phrase-document relevance and keyphrase extraction.

In the future, exploring how pre-trained language models can be adapted to address the challenge of handling long documents in the keyphrase extraction task would be intriguing.

8 Acknowledgments

We thank the three anonymous reviewers for carefully reading our paper and their insightful comments and suggestions. 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

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.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. In *IJCNLP*, pages 543–551.

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.

Boli Chen, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Mosha Chen, and Liping Jing. 2021. Probing {bert} in hyperbolic spaces. In *International Conference on Learning Representations*.

Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. 2020. Hyperbolic interaction model for hierarchical multi-label classification. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 7496–7503. AAAI Press.

Shuyang Dai, Zhe Gan, Yu Cheng, Chenyang Tao, Lawrence Carin, and Jingjing Liu. 2020. Apo-vae: Text generation in hyperbolic space. *CoRR*, abs/2005.00054.

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.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018a. Hyperbolic entailment cones for learning hierarchical embeddings. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1632–1641. PMLR.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. 2018b. Hyperbolic neural networks. In *NeurIPS*, pages 5350–5360.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

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

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *ACL (1)*, pages 1262–1273. The Association for Computer Linguistics.

C. Hopper and B. Andrews. 2011. *The Ricci Flow in Riemannian Geometry*. The Ricci flow in Riemannian geometry :.

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

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL (1)*, pages 3651–3657. Association for Computational Linguistics.

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.

Taeuk Kim, Jihun Choi, Daniel Edmiston, and Sanggoo Lee. 2020. Are pre-trained language models aware of phrases? simple but strong baselines for grammar induction. In *8th International Conference on Learning Representations, ICLR 2020*.

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. Association for Computational Linguistics.

Nathan Linial, Eran London, and Yuri Rabinovich. 1995. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245.

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.

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

Chau Q. Nguyen and Tuoi T. Phan. 2009. An ontology-based approach for key phrase extraction. In *ACL/IJCNLP (Short Papers)*, pages 181–184. The Association for Computer Linguistics.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *NIPS*, pages 6338–6347.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how BERT works. *Trans. Assoc. Comput. Linguistics*, 8:842–866.

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.

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.

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*, 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, and Lin Xiao. 2021. 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.

Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. 2019. Poincare glove: Hyperbolic word embeddings. In *ICLR (Poster)*. OpenReview.net.

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.

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.

Yudong Zhu, Di Zhou, Jinghui Xiao, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Hypertext: Endowing fasttext with hyperbolic geometry. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1166–1171. Association for Computational Linguistics.