2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Hyperbolic Hierarchical Topic-based Keyphrase Generation

Wu Zhuang,Heng Yu, Yafu Li

School of Computer Science and Technology, Civil Aviation University of China, Tianjin, China
2023052082@cauc.edu.cn

Abstract. Keyphrases can concisely describe the high-level topics discussed in a document that usually possesses hierarchical topic structures.Thus, it is crucial to understand the hierarchical topic structuresand employ it to guide the keyphrase identification.However, existing works that integrates the hierarchical topic information into a deep keyphrase generation model still remain in Euclidean space. Their ability to capture the hierarchical structures is limited by the nature of Euclidean space. To this end, we design a new hyperbolic hierarchical topic-based keyphrase generation method(Hyper-HTKG) to effectively exploit the hierarchical topic to improve the keyphrase generationperformance. Concretely, we propose a novel hyperbolic hierarchical topic-guided sequence generation method for keyphrase generation, which consists of two major modules: a hyperbolic hierarchical topic model that learns the latent topic tree across the whole corpus of documents, and a hyperbolic keyphrase generation model to generate keyphrases under hierarchical topic guidance.Finally, these two modules are jointly trained to help them learn complementary information from eachother.To the best of our knowledge, this is the first study to explore ahyperbolic hierarchical topic-based network for keyphrase generation. Compared with seven baseline methods, Hyper-HTKG demonstrates superior performance in experiments conducted on five benchmark datasets.

Keywords: Hyperbolic keyphrase generation,Hyperbolic hierarchical topic model, Hyperbolic keyphrase generation model.

1 Introduction

Keyphrase prediction is the task of automatically generating a set of representative phrases related to the main topic discussed in a given document. Due to their ability to provide advanced topic descriptions for documents, keyphrases are advantageous for many natural language processing (NLP) tasks, such as information extraction [1, 2], text summarization [3], opinion mining [4] and question answering [5].However, the performance of existing methods is still far from being satisfactory. The main reason is that it is very challenging to determine if a phrase or a set of phrases accurately capture the main topics (i.e., salient information) presented in a document.

Automatic keyphrase prediction models can be broadly divided into traditional extraction and deep generation approaches. In particular, traditional extraction methods can only extract presentkeyphrases that appear in a given document, while deep generation methods can generate both present keyphrases as well asabsent keyphrases that do not appear in the given document.

In recent years, some topic-based methods for keyphrase prediction (including extraction and generation) have been proposed, mainly including topic-based extraction methods such as topic-based clustering methods [6, 7] and topical graph-based ranking methods [8, 9, 10, 11, 12, 13]. The work [14] allows the joint learning of the latent flat topic representations.Although these topic-based methods have achieved promising results for the keyphrase prediction task, they all assume that topics are independent of one another and induce topics as flat structures, making generated keyphrases fall into a single topic (i.e., generating duplicate/similar keyphrases). The other work [15] is the first one to leverage the neural hierarchical topics to guide deep keyphrase generation. However, the ability to capture the hierarchical structures is limited by the nature of Euclidean space. Recently, hyperbolic representation methods [16, 17] have been developed to model the latent hierarchical nature of data and demonstrated encouraging results.To effciently utilize hyperbolic embeddings in downstream tasks, researchers have proposed some advanced hyperbolic deep networks, such as hyperbolic neural networks [18] and hyperbolic attention network [19].Though the work [20] is the only hyperbolic topic-based keyphrase generation method, it remains focusing on flat topic representations.

Different from existing deep keyphrase generation approaches which directly encode from a source document and decode to its keyphrases, our proposed method introduces the latent variables to explicitly model underlying hierarchical topics of a source document and to guide the keyphrase generation via collaborative joint training of both the generation model and the hierarchical topic model. This makes our method more effective to capture the semantic hierarchical relations discussed in a document and thus generate keyphrases based on its semantic understanding with good topic coverage and accuracy. To summarize, our main contributions are as follows:
(1)To the best of our knowledge, this is the first attempt to leverage the hierarchical topics to guide deep keyphrase generation in hyperbolic space.
(2) We propose a novel hierarchical topic-guided keyphrase generation model, that not only effectively captures the long and strong dependencies between neighboring target words, but also utilizes the high-level topics discovered for keyphrase generation.
(3)We compare our Hyper-HTKG method with seven seq2seq keyphrase generation methods. Comprehensive experimental results demonstrate that our proposed method outperforms state-of-the-art baseline methods acrossfive publicly-available datasets consistently.

The remaining part of this paper is organized as follows. We first summarize state-of-the-art approaches of keyphrase prediction and text generation in Section 2. Then we introduce the related properties of hyperbolic space in section 3. And the proposed Hyper-HTKG model is presented in Section 4. Finally, we introduce our datasets and experimental results in Section 5, before concluding the paper in Section 6.

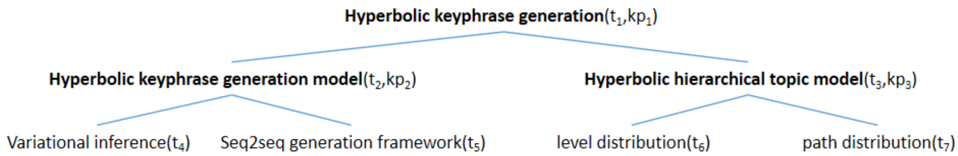

Fig. 1. The hierarchical topic tree structure of this paper.Three high-level topic(t_1, t_2, t_3) description are selected as keyphrases (kp) finally.

2 Related Work

2.1 Topic-based Keyphrase Generation

The traditional extraction methods can be further classified into supervised and unsupervised approaches. In particular, supervised approaches treat keyphrase extraction as a binary classification task, using some classifiers, such as Naïve Bayes classifier [21, 22], boosted decision trees [23] and conditional random fields [24, 25]. In contrast, unsupervised approaches directly treat keyphrase extraction as a ranking problem, scoring each candidate using different kinds of unsupervised learning techniques, such as clustering [6, 7] and graph-based ranking [8, 9, 10, 11, 12, 26].

Topic information is used mainly in graph-based methods and most attempts involve biasing the ranking function towards topic distribution. Existing graph-based methods incorporating topic information induced by latent Dirichlet allocation (LDA) [27] include TopicalPageRank [10], cTPR [12], TPR [28], MIKE [11] and SalienceRank [13]. The other two works [8, 9] represent a given document as a multipartite graph of both topics and keyphrase candidates, and then select keyphrases from the top-ranked candidates, in which topics are defined as clusters of similar candidates. Nevertheless, in all these topic-based extraction methods, topics are independent of one another and organized as flat structures. Inaddition, compared with the newly developed generation methods, the traditional approaches suffer from poor performance [29].

2.2 Deep Keyphrase Generation

CopyRNN [29] is the first to employ the attentional sequence to sequence (seq2seq) framework [30] with the copying mechanism [31] to generate both present and absent keyphrases for a document. Following this work, numerous extensions have been proposed to boost its generation ability. For instance, some studies incorporate different types of side information into seq2seq neural networks to improve keyphrase generation, such as correlation among keyphrases [32], title of source document [33], syntactic constraints [34] and topic information [35]. In addition, Ye et al., [36] propose a semi-supervised keyphrase generation model that utilizes both abundant unlabeled data and limited labeled data.

The above-mentioned early methods which use the standard seq2seq network can not generate multiple keyphrases and determine the appropriate number of keyphrases at a time for a target document. To overcome this drawback, Yuan et al., [37] introduce a new One2Seq training paradigm in the seq2seq network to generatemultiple keyphrases and decide the suitable number of keyphrases for a target document. Ye et al., [38] propose a One2Set paradigm to predict the keyphrases as a set, which eliminates the bias caused by the predefined order in One2Seq paradigm [37]. In addition, some works focus on improving the decoding process of seq2seq networks. Chen et al., [39] propose an exclusive hierarchical decoding framework and use either a soft or a hard exclusion mechanism to reduce duplication. Ahmad et al., [40] introduce an extractor-generator in the decoding to jointly extract and generate keyphrases from a target document. Bahuleyan et al., [41] adopt neural an unlikelihood objective to avoid generating duplicate keyphrases.

Besides the seq2seq networks (which can be implemented by the long short-term memory (LSTM) [42] or gated recurrent units (GRU) [43]), the neural graph-based networks, that extend traditional graph-based keyphrase ranking, have been used in keyphrase generation. Prasad et al., [44] firstly combine the advantages of traditional graph-based ranking methods and recent neural network-based approaches. Specifically, this method incorporates the global information (i.e., TextRank ranking scores) into a graph attention network (GAT) [45] to extract keyphrases. Sun et al., [46] employ a graph convolutional neural network (GCN) [47] to encode the word graph into the corresponding representations and then adopt a pointer network [48] with diversity enabled attentions to generate keyphrases. Subsequently, Kim et al., [49] extend the word graph with structure information, and use GCN to extract the keyphrases for Web documents. Ye et al., [50] also enrich the word graph with related references and employ a GAT to generate the keyphrases.

We observe that almost all the existing deep keyphrase prediction approaches do not consider integrating the latent hierarchical topic information into the seq2seq framework to improve keyphrase prediction. In this paper, we first incorporate the hierarchical topical information into the sequence generation model, which can ensure that the generated keyphrases cover comprehensive topics and thus provide high-level topic description.

2.3 Hyperbolic Representation

An increasing number of research has shown that many types of complex data exhibit non-Euclidean structures [51]. Recently, hyperbolic embedding methods have been proposed to learn the latent representation of hierarchical data and demonstrated encouraging results. In the field of NLP, hyperbolic representation learning has been successfully applied to generating word embeddings [52] and sentence representations [53], and inferring concept hierarchies from large text corpora [54]. In addition, hyperbolic geometry has been integrated into recent advanced hyperbolic deep learning frameworks, such as hyperbolic neural networks [18], and hyperbolic attention network [19].

3 Preliminaries

Hyperbolic space, specifically referring to a simply connected manifolds with constant negative curvature [55], can be thought of as a continuous analogue of tree and is more suitable for learning data with hierarchical structures. The hyperbolic space can be constructed using various isomorphic models (i.e., these models can be converted into each other). In this paper, we follow the majority of NLP works and employ the Poincaréball model with the curvature set as -1, whose distance function is differentiable.

Poincaré ball model The n-dimensional *Poincaré ball model* $P^n = (B^n, g^P)$ is defined by a Riemannian manifold $B^n = \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\| < 1\}$ with the metric tensor $g^P(\mathbf{x}) = (\frac{2}{1-\|\mathbf{x}\|^2})^2 g^\epsilon$, where $\|\cdot\|$ denotes the Euclidean norm, and $g^\epsilon = \mathbf{I}_n$ is the Euclidean metric tensor. The *induced distance* between two points $\mathbf{x}, \mathbf{y} \in P^n$ is defined as

$$d_p(\mathbf{x}, \mathbf{y}) = \cosh^{-1}(1 + \frac{2\|\mathbf{x}-\mathbf{y}\|^2}{(1-\|\mathbf{x}\|^2)(1-\|\mathbf{y}\|^2)}) \tag{1}$$

where$\cosh^{-1}(x) = \ln(x + \sqrt{x^2 - 1})$ is an inverse hyperbolic cosine function.

The induced distance can place root node near the center of the ball and leaf nodes near the boundary of the ball to ensure that the distance from the root node to each of leaf nodes is relatively small while the distance between leaf nodes is relatively large. This explains why hyperbolic space can be seen as a tree-like hierarchical structure.

Klein model To define the *hyperbolic average*, we employ the Klein model of hyperbolic space. The n-dimensional *Klein model*$K^n = (B^n, g^K)$ is also defined in a manifold B^n with the different metric tensor g^K. The Poincaré model and Klein model describe the same hyperbolic space using different coordinates. Thus, these two models can be converted into each other. Given a point $\mathbf{x}^P \in P^n$ in the Poincaré ball model, we convert it to the Klein model by

$$\mathbf{x}^k = \frac{2\mathbf{x}^p}{1+\|\mathbf{x}^p\|^2} \tag{2}$$

Similarly, a point $\mathbf{x}^k \in k^n$ in Klein model can be converted into Poincaré ball model as

$$\mathbf{x}^p = \frac{\mathbf{x}^k}{1+\sqrt{1-\|\mathbf{x}^k\|^2}} \tag{3}$$

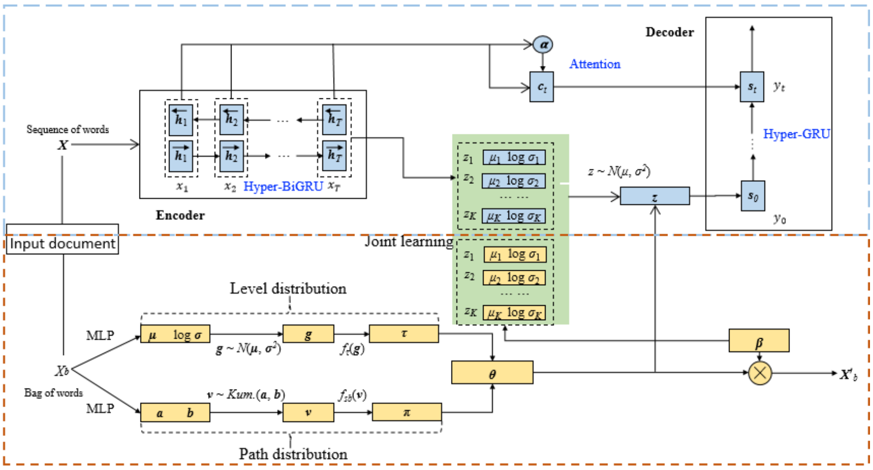

Fig. 2. The overall architecture of the proposed Hyper-HTKG model.

4 Methodology

4.1 Problem Definition and Framework

Given a corpus of documents $D = \{d_i\}_{i=1}^{|D|}$, where each document d ($d \in D$) is treated as a sequence of words $X = (x_1, \ldots, x_{T_d})$ with length T_d, the goal of a keyphrase generation method is to find a model to generate a set of keyphrases $K = \{p_j\}_{j=1}^{|K|}$ for document , where each keyphrase p can be treated as a sequence of words $Y = (y_1, \ldots, y_{|p|})$. Note that as in existing deep text generation models, we use X and Y to denote the word sequence of an input document and the word sequence of its keyphrase, respectively.

The overall architecture of our proposed method is shown in Figure 2. It consists of two main modules: (1) a hyperbolic hierarchical topic model that computes a topic distribution over a tree for each word occurrence in a corpus, and (2) a hyperbolic keyphrase generation model to generate keyphrases with designated topic guidance. We jointly train them with an inconsistency loss so that they can learn complementary information from each other accurately. Below we first introduce the two main modules and then describe how they are jointly trained in detail.

4.2 Hyperbolic Hierarchical Topic Model

One innovation of this study is that it incorporates hierarchical topical information into keyphrase generation explicitly. Based on the current development of topic modeling, we follow the spirit of the hierarchical topic models [56] and adapt it to discover latent hierarchical topics. In this subsection, we first introduce the technical background and preliminaries and then describe the details of this model.

4.2.1 Technical Background and Preliminaries

Constructing a topic tree involves mainly two aspects: how to infer the latent topics in the text corpus, and how to organize these topics into a hierarchy. Traditional hierarchical topic models such as HLDA[59], nHDP [60] and rCRP [61], use conventional inference algorithms such as collapsed Gibbs sampling [62] and mean-field approximation [63], to infer the latent hierarchical topics. Current hierarchical topic models TSNTM [57], HTV [58] and nTSNTM [56], leverage the autoencoder variational Bayes framework, which can be trained together with networks and therefore has better adaptability and scales to large datasets.

To construct a topic tree with an infinite number of branches and levels, the existing methods follow the classical hierarchical LDA model nCRP [59, 63], which draws the path distribution from a nested stick-breaking construction as followings

$$v_k \sim Beta(1, \gamma), \pi_k = \pi_{par}(k) v_k \prod_{j=1}^{k-1}(1 - v_j) \tag{4}$$

and draws the level distribution from a stick-breaking construction as followings

$$\eta_l \sim Beta(1, \alpha), \theta_l = \eta_l \prod_{j=1}^{l-1}(1 - \eta_j) \tag{5}$$

where$k \in \{1, ..., K\}$ and *par(k)* denote respectively the *k*-th topic and its parent. $l \in \{1, ..., L\}$ denotes the *l*-th level. v_k and η_l are stick proportions of topic *k* and level *l*, respectively

4.2.2 Generative Process

Given a document *d*, we process it into a bag-of-words vector $X_b \in \mathbb{Z}_+^{|V|}$, with $\mathbb{Z}_+$ denoting non-negative integers and *V* representing the vocabulary, in which each element reflects the number of times the corresponding word occurs in thedocument. To sample a topic for a word x_n in document d, a path c_n from the root to a leaf node and a level l_n are drawn. Let $\boldsymbol{\beta}_{c_n}, l_n$be the topic distribution of the topic in path c_n and at level l_n. The full generative process of each word is given as follows
1. For a document *d*,
Draw a breaking proportions: $\boldsymbol{v}_d \sim$ Beta(α_0, β_0)
Obtain a path distribution: $\boldsymbol{\pi}_d = f_{sb}(v_d)$
Draw a Gaussian vector: $\boldsymbol{g}_d \sim N\ (0, \boldsymbol{I}^2\ )$
Obtain a level distribution: $\boldsymbol{\tau}_d = f_\tau(\boldsymbol{g}_d)$
2. For a word x_n in document d,
Draw a path: $c_n \sim$Mult$(\boldsymbol{\pi}_d\ )$, for $n \in [1, N_d\ ]$
Draw a level: $l_n \sim$Mult$(\boldsymbol{\tau}_d\ )$, for $n \in [1, N_d\ ]$
Draw a word: $x_n \sim$Mult$(\boldsymbol{\beta}_{c_n}, l_n)$, for $n \in [1, N_d\ ]$
where $f_{sb}(\cdot)$ is a stick-breaking construction function, and $f_\tau(\cdot)$ is a neural perceptron with softmax activation to transform a Gaussian sample to a level distribution.

4.3 Hyperbolic Keyphrase Generation Guided by Hierarchical Topic

Different from traditional seq2seq keyphrase generation methodssuch as CopyRNN [67] and SEG-Net [40], our keyphrase generation model is a sequence generation model, based on the seq2seq framework model in hyperbolic space. Specifically, we introduce a latent variable, which is guided by the hierarchical topic model described in the previous section, to model the underlying topic space as a global signal for keyphrase generation. Thus, it should be able to capture the high-level topic in a given document.

4.3.1 Hyperbolic Encoder

This module aims at encoding an input document into continuous vectors. Let X= (x_1, ..., x_T) be a sequence of words within an input document, and x= (x_1, ..., x_T)be its corresponding sequence of word embeddings. We adopt a bidirectional gated recurrent unit (Hyper-BiGRU) [71] as the encoder, which maps the input word sequence X into a set of contextualized hidden states $\boldsymbol{h}$= ($\boldsymbol{h_1}$, ..., $\boldsymbol{h}_T$) as

$$\boldsymbol{h_1},\ldots,\boldsymbol{h}_T = \text{Hyper} - \text{BiGRU}(x_1,\ldots,x_T) \tag{6}$$

In this way, each contextualized vector $\boldsymbol{h_i}$ encodes informationabout the i-th word with respect to all the other surrounding words in the sequence. The last hidden state of the encoder $\boldsymbol{h}_T$ is used to calculate the latent topic variable **z**.

4.3.2 Hierarchical Topic-guided Gaussian Mixture Posterior

In this model, we consider incorporating the topic information into latent variables. Each topic is drawn from a topic-dependent multivariate Gaussian distribution, computed as

$$p(z|\boldsymbol{X}) = \sum_{k=1}^{K} \theta_k(\boldsymbol{X}_b)\, N(\hat{z}_k; \widehat{\boldsymbol{\mu}}_k(\boldsymbol{X}), \widehat{\boldsymbol{\sigma}}_k^2(\boldsymbol{X})) \tag{7}$$

where θ_k is the usage of topic k in a document, computed by our Hyperbolic HTM model. To estimate $\hat{z}_k$, we introduce the fully connected layer to obtain vectors $\widehat{\boldsymbol{\mu}}_k$and $\log \widehat{\boldsymbol{\sigma}}_k$as follows

$$\begin{aligned} \widehat{\boldsymbol{\mu}}_k &= W_{\boldsymbol{\mu}_k}\boldsymbol{h}_T + \boldsymbol{b}_{\boldsymbol{\mu}_k} \\ \log \widehat{\boldsymbol{\sigma}}_k &= W_{\sigma_k}\boldsymbol{h}_T + \boldsymbol{b}_{\sigma_k} \end{aligned} \tag{8}$$

Finally, to obtain a representation for the latent topic variable **z**,we follow the reparameterization trick ofVAE to implement it.

4.3.3 Hyperbolic Decoder

Given a source document $\boldsymbol{X}$ and a continuous latent topic variable $\boldsymbol{z}$, the process to generate its keyphrase $\boldsymbol{Y}$ is defined as following conditional probability

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{t=1}^{|\boldsymbol{Y}|} p(y_t|\boldsymbol{Y}_{<t}, \boldsymbol{z}, \boldsymbol{X}) p(\boldsymbol{z}|\boldsymbol{X}) \tag{9}$$

where $\boldsymbol{Y}_{<t} = (y_1, \ldots, y_{t-1})$ is a previously generated wordsequence.The decoder is another forward GRU, which is used to generate the sequence of keyphrases by predicting the next word $y_t$based on the hidden state s_t of the decoder at timestep *t*. Both y_t ands_t are conditioned on y_{t-1} and c_t of the input sequence. Formally, the hidden state s_t and decoding function can be written as

$$\begin{gathered} s_t = \mathrm{GRU}_f(y_{t-1}, s_{t-1}, c_t) \\ p(y_t|\boldsymbol{Y}_{<t}, \boldsymbol{z}, \boldsymbol{X}) = g(y_{t-1}, s_t, c_t) \end{gathered} \tag{10}$$

where $c_t = \sum_i \alpha_{ti} \boldsymbol{h}_i$ is a source context vector computed as the weighted sum of the source hidden states $\{\boldsymbol{h}_i\}$ using the attention mechanism [71], and $g(\cdot)$ is a nonlinear multi-layered function that outputs the probability of y_t. The latent topic variable $\mathbf{z}$ is used to initialize the hidden state s_0 in the decoder.

Finally, we minimize the cross entropy loss function to train this generation model

$$\boldsymbol{\mathcal{L}}_{kg} = -\sum_{i=1}^{N} \log(p(y_i|\boldsymbol{X}, \boldsymbol{Y}_{<i}, \boldsymbol{z})) \tag{11}$$

where *N* denotes the length of target keyphrases, and $\boldsymbol{z}$ is the latenttopic of the given document.

4.4 Joint Learning

Since keyphrase generation and topic modeling both aim to distill salient information from input documents, we jointly train the two modules to help them learn complementary information from each other. The loss function of our model consists of three parts. Two of them, namely, hierarchical topic loss $\boldsymbol{\mathcal{L}}_{ht}$ and keyphrase generation loss $\boldsymbol{\mathcal{L}}_{kg}$, have been given in the previous subsections.

To push the hierarchical topic-guided Gaussian mixture computed in VNKG towards the corresponding distribution computed in Hyperbolic HTM, we devise the third loss — an inconsistency loss $\boldsymbol{\mathcal{L}}_{ic}$. For these two mixture Gaussian distributions $p(\boldsymbol{z}|\boldsymbol{X}_b) = \sum_{i=1}^{K} \theta_i N(\mu_i, \sigma_i^2)$and $p(\boldsymbol{z}|\boldsymbol{X}) = \sum_{i=1}^{K} \theta_i N(\hat{\mu}_i, \hat{\sigma}_i^2)$, their Kullback–Leibler (KL) divergence is upper-bounded by

$$\boldsymbol{\mathcal{L}}_{ic} = \mathrm{KL}(p(\boldsymbol{z}|\boldsymbol{X}_b))||p(\boldsymbol{z}|\boldsymbol{X}) \le \mathrm{KL}(\boldsymbol{\theta}||\boldsymbol{\theta}) + \sum_{i=1}^{K} \theta_i \mathrm{KL}(N(\mu_i, \sigma_i^2)||N(\hat{\mu}_i, \hat{\sigma}_i^2)) \tag{12}$$

where $\mathrm{KL}(\boldsymbol{\theta}||\boldsymbol{\theta})$ is equal to 0. The general form of this formula hasbeen proven to be correct in the study [72].

The final overall loss of the entire framework's training objective is the linear combination of the three parts, defined as

$$\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{L}}_{ht} + \boldsymbol{\mathcal{L}}_{kg} + \boldsymbol{\mathcal{L}}_{ic} \tag{13}$$

5 Experiments

5.1 Datasets

We employ the dataset KP20k collected by Meng et al., [67], which contains a large amount of high-quality scientific metadata in the computer science domain from various online digital libraries. In this dataset, each example contains a title and an abstract of a scientific publication as source text, and multiple author-assigned keywords as target keyphrases. Following previous works [37, 67], we split this dataset into training, validation and test sets, and use the training set to train all the deep seq2seq models. We use the validation set to find the optimal hyperparameters during the training process. Finally, we apply our models in the test set and report their performance.

In order to evaluate the proposed model comprehensively, we also test the model trained with KP20k on other four widely-adopted public datasets from the scientific domain, namely, Inspec [73], Krapivin [74], SemEval-2010 [75] and NUS [76]. The detailed statistic information of the above five datasets are shown in Table 1, along with the number of documents (#Docs), the number and the percentage of present keyphrases (#PKps and %PKps), the number and the percentage of absent keyphrases (#AKps and %AKps), and the average number of keyphrases per document (#Avg.Kps).

Table 1. Summary of the training, validating and testing datasets.

Dataset		#Docs	#PKps	%PKps	#AKps	%AKps	#Avg.Kps
Train	KP20k	509986	1696532	63.0	995744	37.0	5.28
Valid	KP20k	20000	66131	62.9	39041	37.1	5.26
Test	KP20k	20000	66441	62.9	39119	37.1	5.28
	Inspec	500	3602	73.6	1293	26.4	9.79
	Krapivin	400	1297	55.6	1037	44.4	5.84
	NUS	211	1191	52.2	1088	47.8	10.8
	SemEval	100	612	42.4	831	57.6	14.43

5.2 Comparative Methods

To comprehensively evaluate the performance of our Hyper-HTKG1 , we compare our method with seven current deep seq2seq generation baselines on five benchmark datasets as follows:

(1) CopyRNN [29] is the first to use seq2seq network to generate keyphrases. Here, we replace it with CopyRNN+ which is re-implemented CopyRNN with best results [78].

(2) CopyCNN [84] applies a convolutional neural network-based encoder-decoder framework to generate keyphrases.

(3) KG-KE-KR-M [85] is a multi-task learning method using extractive and generative models to generate keyphrases.

(4) CatSeq [37] has the same framework as CopyRNN, with the key difference in training paradigm.

(5) CatSeqTG-2RFl [86] is a simple extension of CatSeq using reinforcement learning to generate both sufficient and accurate keyphrases.

(6) ExHiRD-h [39] uses an exclusive hierarchical decoder to avoid generating duplicated keyphrases.

(7) One2Set [38] is a new training paradigm without predefining an order to concatenate the keyphrases.

5.3 Evaluation Metrics

For fairly comparing different approaches, we follow the literature and adopt top-N macro-averaged precision, recall and F1-measure as the evaluation metrics. In particular, precision is defined as the number of correctly predicted keyphrases over the number of all predicted keyphrases, recall is defined as the number of correctly predicted keyphrases over the total number of data records, and F1 is the harmonic mean of precision and recall.

Note F1@k is used in almost all existing works on the keyphrase extraction and generation, in which k is a fixed number of top-N predictions. F1@O is recently proposed in the work [37] as one of our evaluation metrics, in which O is the number of author-assigned keyphrases. This means that the number of predicted phrases taken for evaluation is the same as the number of ground truth keyphrases for each document.

5.4 Experimental Setup

We follow the previous works [29, 37] to pre-process the experimental data, including lowercasing, tokenizing, etc. Particularly, the top 50,000 and 10,000 most frequently-occurred words in the training data are selected as the vocabulary shared in the sequence encoder and decoder, and as the bag-of-words vocabulary in the neural hierarchical topic model, respectively.

For the hyperbolic hierarchical topic model, we set the size of hidden layers to 256. The parameters α_0 and β_0 for the topic-word distribution are empirically set to 1 and 10, respectively. For the neural keyphrase generation model, the word embeddings are initialized first using normal distribution by the method, and the size of word embedding is set as 150. The size of hidden state of hyperbolic Bi-GRU encoder is set as 150, and the size of hidden state of forward GRU decoder is set as 300.

In the training process, we adopt One2One training paradigm [67]and use Adam as optimizer to optimize all the parameters. The initial learning rate is set as 0.001 and the gradient clipping is set as 1. The batch sizes of the topic model and the keyphrase generation model are set to 1024 and 128, respectively. We halve the learning rate when the validation performance drops, and stop training if it does not improve for three successive iterations. In addition, we pre-train the hierarchical topic model for 100 epochs before thejoint training as the convergence speed of our hyperbolic keyphrase generation model is much faster than our hyperbolic hierarchical topic model. We employ the simple KL cost annealing technique. More specifically, we add a variable weight to the

KL term in the loss function at training time. At the start of training, we set that weight to 0, and then we gradually increase this weight to 1 as the training progresses. In the testing process, our models use the beam search with a width of 200 and a max depth of6.

Table 2. Present keyphrase prediction results on five datasets. The best performing score is highlighted in bold and the second best score is highlighted with underline.

Model	KP20k		Inspec		Krapivin		NUS		SemEval	
	F1@5	F1@O	F1@5	F1@O	F1@5	F1@O	F1@5	F1@O	F1@5	F1@O
CopyRNN	31.7	33.5	24.4	29.0	30.5	32.5	37.6	<u>40.6</u>	31.8	**31.7**
CopyCNN	35.1	-	28.5	-	31.4	-	34.2	-	29.5	-
KG-KE-KR-M	31.7	**38.8**	25.7	<u>31.4</u>	27.2	31.7	28.9	38.4	20.2	30.3
CatSeq	31.4	31.9	<u>29.0</u>	30.7	30.7	32.4	35.9	38.3	30.2	<u>31.0</u>
CatSeqTG	32.6	35.7	26.6	22.4	31.2	**34.7**	36.5	39.6	27.7	25.5
Exhird-h	31.1	<u>37.4</u>	25.3	28.9	28.4	30.6	-	-	29.2	26.6
One2Set	<u>35.5</u>	36.9	28.2	25.4	<u>31.5</u>	<u>34.3</u>	<u>39.7</u>	39.5	**34.0**	30.2
Hyper-HTKG	**37.4**	**38.8**	**30.6**	**31.7**	**31.7**	32.3	**40.9**	**41.7**	<u>32.0</u>	30.5

Table 3. Absent keyphrase prediction results on five datasets. The best performing score is highlighted in bold and the second best score is highlighted with underline.

Model	KP20k		Inspec		Krapivin		NUS		SemEval	
	F1@5	F1@O	F1@5	F1@O	F1@5	F1@O	F1@5	F1@O	F1@5	F1@O
CopyRNN	3.23	3.97	1.44	1.23	5.14	<u>5.54</u>	3.35	3.25	2.05	<u>2.20</u>
CatSeq	1.50	2.51	0.40	0.35	1.80	1.67	1.60	1.22	1.60	1.44
CatSeqTG	2.78	2.16	1.12	0.82	2.97	1.88	2.46	2.10	2.04	1.78
Exhird-h	1.57	3.22	1.03	1.52	2.06	2.49	-	-	1.51	1.14
One2Set	<u>3.70</u>	<u>5.02</u>	<u>2.10</u>	**1.63**	**5.53**	3.43	<u>4.02</u>	<u>3.61</u>	<u>2.42</u>	2.02
Hyper-HTKG	**4.20**	**5.27**	**2.24**	<u>1.59</u>	<u>5.41</u>	**5.82**	**4.96**	**4.82**	**2.99**	**2.67**

5.5 Performance Comparison

We compare Hyper-HTKG with the baselines on five datasets, and the experimental results for present and absent keyphrase prediction are shown in Table 2 and 3, respectively. Due to space limitations and metric specialization, we present only results obtained with the most suitable metrics for each type of methods. Specifically, we choose F1@5 andF1@O for the present and absent methods.

5.5.1 Present keyphrase prediction

From the results of predicting present keyphrases illustrated in Table 2, we can see that our Hyper-HTKG outperforms all the seq2seq baseline methods by significant margins in three out of five datasets (including KP20k, Inspec and NUS) in terms of all the metrics. Specifically, Hyper-HTKG achieves the improvement of 1.9F1@5 points points on KP20k over the best baselines, of 1.6 F1@5 pointsand 0.3F1@O points on Inspec, and of 1.2F1@5 points and 1.1F1@O points on NUS, respectively.

These results illustrate Hyper-HTKG can achieve the average increase of 2.1 points on these metrics, which is a significant improvement in the current keyphrase prediction task. On both Krapivin and SemEval datasets, Hyper-HTKG performsslightly worse than the best baselines. This slight performance drop may be caused by the various topics discussed in the given datasets. For example, the selected articles in SemEval dataset belong to both computer science and economics domains.

5.5.2 Absent keyphrase prediction

Unlike present keyphrases, absent keyphrases do not appear in the target document, and thus predicting them is very challenging and requires comprehensive understanding the latent document semantic. From the results of predicting absent keyphrases presented in Table 3, we can see that Hyper-HTKG substantially outperforms the baselines according to all the metrics, and correctly generates more absent keyphrases than the baselines on the five datasets, especially on KP20k (0.5F1@5 points and 0.25 F1@O points over the best existing methods), NUS (0.94F1@5 points and 1.21 F1@O points) andSemEval (0.57F1@5 points and 0.65 F1@O points). Overall, the absent keyphrase prediction results indicate that Hyper-HTKG is capable of understanding the underlying document semantic better than all the baselines, and thus generating much better results.

Table 4. Ablation study on five public datasets.

	Model	KP20k		Inspec		Krapivin		NUS		SemEval	
		F1@5	F1@O	F1@5	F1@O	F1@5	F1@O	F1@5	F1@O	F1@5	F1@O
present	Hyper-HTKG	37.4	38.8	30.6	31.7	31.7	32.3	40.9	41.7	32.0	30.5
	w/o HT	34.1	36.3	27.4	28.0	26.9	28.4	34.6	36.2	28.2	27.5
	w/o TG	33.0	34.3	25.5	27.6	26.1	27.8	32.4	33.6	27.1	26.4
absent	Hyper-HTKG	4.20	5.27	2.24	1.59	5.41	5.82	4.96	4.82	2.99	2.67
	w/o HT	2.73	3.32	1.86	1.44	2.89	3.10	3.27	3.62	1.98	2.02
	w/o TG	1.99	2.20	1.46	1.27	2.05	2.26	2.07	2.42	1.69	1.80

5.6 Ablation Study

To analyze the relative contributions of different components to the model performance in predicting present and absent keyphrases, we compare our full model Hyper-HTKG with the following ablated variants: (1) w/o HT(hierarchical topic) where the hierarchical topic model is replaced by the flat topic model NTM, (2) w/o TG(topic guidance), where we directly use the hyperbolic keyphrase generation model.

From the results shown in Table 4, we have the following observations: (1) Replacing the hierarchical topics with the flat topics leads to performance drops on all datasets, indicating that the hierarchical topic is effective information to improve keyphrase generation. (2) The simple concatenation results in significant performance drop on all datasets. However, compared to the earliest baseline CopyRNN, slight improvements of the performance are observed in conjunction with the results shown in Table 2 and Table 3. These results indicate that NTM can effectively leverage the topic information to guide the keyphrase generation.

6 Conclusion

In this study, we propose a hyperbolic hierarchical topic-guided keyphrase generation method, which incorporates the hierarchical topic information into keyphrase generation explicitly. In particular, we jointly learn both latent hierarchical topics and keyphrases, allowing our model to better exploit the mutual reinforcement between them, and accurately capturing the topics and relations between them discussed in a given document. We conducted comprehensive experiments to demonstrate its advantages and effectiveness. In future, we plan to evaluate Hyper-HTKG on a large corpus with comprehensive coverage of diverse topics.

References

1. Xiaojun Wan and Jianguo Xiao. 2008. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In Proceedings of AAAI.
2. Tianchi Yang, Linmei Hu, Chuan Shi, Houye Ji, Xiaoli Li, and Liqiang Nie. 2021. HGAT: Heterogeneous graph attention networks for semi-supervised short text classification. ACM Transactions on Information Systems (TOIS) 39, 3 (2021), 1–29.
3. Lu Wang and Claire Cardie. 2013. Domain-Independent Abstract Generation for Focused Meeting Summarization. In Proceedings of ACL.
4. Gabor Berend. 2011. Opinion Expression Mining by Exploiting Keyphrase Extraction. In Proceedings of IJCNLP.
5. Yixuan Tang, Weilong Huang, Qi Liu, and Beibei Zhang. 2017. QALink: Enriching text documents with relevant Q&A site contents. In Proceedings of CIKM.
6. Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In Proceedings of WWW.
7. Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In Proceedings of EMNLP.

8. Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite graphs. In Proceedings of NAACL.
9. Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph based topic ranking for keyphrase extraction. In Proceedings of IJCNLP.
10. Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In Proceedings of EMNLP.
11. Yuxiang Zhang, Yaocheng Chang, Xiaoqing Liu, Sujatha Das Gollapalli, Xiaoli Li, and Chunjing Xiao. 2017. Mike: keyphrase extraction by integrating multidimensional information. In Proceedings of CIKM.
12. Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achanauparp, EePeng Lim, and Xiaoming Li. 2011. Topical keyphrase extraction from twitter. In Proceedings of ACL.
13. Nedelina Teneva and Weiwei Cheng. 2017. Salience rank: Efficient keyphrase extraction with topic modeling. In Proceedings of ACL (Short Papers)
14. Yue Wang, Jing Li, Hou Pong Chan, Irwin King, Michael R. Lyu, and Shuming Shi. 2019. Topic-Aware Neural Keyphrase Generation for Social Media Language. In Proceedings of ACL.
15. Tianchi Yang, Linmei Hu, Chuan Shi, Houye Ji, Xiaoli Li, and Liqiang Nie. 2021. HGAT: Heterogeneous graph attention networks for semi-supervised short text classification. ACM Transactions on Information Systems (TOIS) 39, 3 (2021), 1–29.
16. Zhang, Y., Jiang, T., Yang, T., Li, X., & Wang, S. (2022, July). Htkg: Deep keyphrase generation with neural hierarchical topic guidance. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1044-1054).
17. Nickel, M., Kiela, D. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In: Proceedings of ICML.
18. Nickel, M., Kiela, D. 2017. Poincare embeddings for learning hierarchical representations. In: ´Proceedings of NIPS.
19. Gulcehre, C., Denil, M., Malinowski, M., Razavi, A., Pascanu, R., Hermann, K.M., Battaglia, P., Bapst, V., et al. 2019. Hyperbolic attention networks. In: Proceedings of ICML.
20. Zhang, Y., Yang, T., Jiang, T., Li, X., & Wang, S. 2022. Hyperbolic deep keyphrase generation. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 521-536). Cham: Springer International Publishing.
21. Olena Medelyan, Eibe Frank, and Ian H Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In Proceedings of EMNLP.
22. Eibe Frank, Gordon W Paynter, Ian H Witten, Carl Gutwin, and Craig G Nevill Manning. 1999. Domain-specific keyphrase extraction. In Proceedings of IJCAI.
23. Lucas Sterckx, Thomas Demeester, Chris Develder, and Cornelia Caragea. 2016. Supervised keyphrase extraction as positive unlabeled learning. In Proceedings of EMNLP
24. Rabah Alzaidy, Cornelia Caragea, and C Lee Giles. 2019. Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents. In Proceedings of WWW.
25. Sujatha Das Gollapalli, Xiao-Li Li, and Peng Yang. 2017. Incorporating Expert Knowledge into Keyphrase Extraction. In Proceedings of AAAI.
26. Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In Proceedings of EMNLP.
27. David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. Journal of machine Learning research 3 (2003), 993–1022.
28. Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2015. Topical word importance for fast keyphrase extraction. In Proceedings of WWW.

29. Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep Keyphrase Generation. In Proceedings of ACL.
30. Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In Proceedings of NIPS.
31. Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In Proceedings of ACL.
32. Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. Keyphrase Generation with Correlation Constraints. In Proceedings of EMNLP.
33. Wang Chen, Yifan Gao, Jiani Zhang, Irwin King, and Michael R Lyu. 2019. TitleGuided Encoding for Keyphrase Generation. In Proceedings of AAAI.
34. Jing Zhao and Yuxiang Zhang. 2019. Incorporating Linguistic Constraints into Keyphrase Generation. In Proceedings of ACL
35. Yue Wang, Jing Li, Hou Pong Chan, Irwin King, Michael R. Lyu, and Shuming Shi. 2019. Topic-Aware Neural Keyphrase Generation for Social Media Language. In Proceedings of ACL.
36. Hai Ye and Lu Wang. 2018. Semi-Supervised Learning for Neural Keyphrase Generation. In Proceedings of EMNLP.
37. Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Peter Brusilovsky, Daqing He, and Adam Trischler. 2020. One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases. In Proceedings of ACL.
38. Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. ONE2SET: Generating Diverse Keyphrases as a Set. In Proceedings of ACL.
39. Wang Chen, Hou Pong Chan, Piji Li, and Irwin King. 2020. Exclusive Hierarchical Decoding for Deep Keyphrase Generation. In Proceedings of ACL.
40. Wasi Uddin Ahmad, Xiao Bai, Soomin Lee, and Kai-Wei Chang. 2021. Select, Extract and Generate: Neural Keyphrase Generation with Layer-wise Coverage Attention. In Proceedings of ACL.
41. Hareesh Bahuleyan and Layla El Asri. 2020. Diverse keyphrase generation with neural unlikelihood training. In Proceedings of COLING.
42. Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 1735–1780.
43. Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of EMNLP.
44. Animesh Prasad and Min-Yen Kan. 2019. Glocal: Incorporating Global Information in Local Convolution for Keyphrase Extraction. In Proceedings of NAACL.
45. Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. In Proceedings of ICRL.
46. Zhiqing Sun, Jian Tang, Pan Du, Zhi-Hong Deng, and Jian-Yun Nie. 2019. Divgraphpointer: A graph pointer network for extracting diverse keyphrases. In Proceedings of SIGIR.
47. Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In Proceedings of ICLR.
48. Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. Advances in neural information processing systems.
49. Jihyuk Kim, Young-In Song, and Seung-won Hwang. 2021. Web Document Encoding for Structure-Aware Keyphrase Extraction. In Proceedings of SIGIR.
50. Jiacheng Ye, Ruijian Cai, Tao Gui, and Qi Zhang. 2021. Heterogeneous Graph Neural Networks for Keyphrase Generation. In Proceedings of EMNLP

51. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. IEEE Signal Process. Mag. 34(4), 18–42 (2017)
52. Tifrea, A., Becigneul, G., Ganea, O.E.2019. Poincaré glove: Hyperbolic word embeddings. In: Proceedings of ICLR.
53. Dhingra, B., Shallue, C.J., Norouzi, M., Dai, A.M., Dahl, G.E.: Embedding text in hyperbolic spaces. In: Proceedings of Twelfth Workshop on TextGraphs (2018)
54. Le, M., Roller, S., Papaxanthos, L., Kiela, D., Nickel, M.: Inferring concept hierarchies from text corpora via hyperbolic embeddings. In: Proceedings of ACL (2019)
55. Birman, G.S., Ungar, A.A.2001. The hyperbolic derivative in the poincare ball model of hyperbolic geometry. Journal of mathematical analysis and applications 254(1), 321–333.
56. Ziye Chen, Cheng Ding, Zusheng Zhang, Yanghui Rao, and Haoran Xie. 2021. Tree-structured topic modeling with nonparametric neural variational inference. In Proceedings of ACL.
57. Masaru Isonuma, Junichiro Mori, Danushka Bollegala, Ichiro Sakata, et al. 2020. Tree-Structured Neural Topic Model. In Proceedings of ACL.
58. Dang Pham and Tuan Le. 2021. Neural Topic Models for Hierarchical Topic Detection and Visualization. In Proceedings of ECML-PKDD
59. David M Blei, Thomas L Griffiths, Michael I Jordan, Joshua B Tenenbaum, et al. 2003. Hierarchical topic models and the nested Chinese restaurant process.. In Proceedings of NIPS.
60. John Paisley, Chong Wang, David M Blei, and Michael I Jordan. 2014. Nested hierarchical Dirichlet processes. IEEE transactions on pattern analysis and machine intelligence 37, 2 (2014), 256–270.
61. Joon Hee Kim, Dongwoo Kim, Suin Kim, and Alice Oh. 2012. Modeling topic hierarchies with the recursive chinese restaurant process. In Proceedings of CIKM.
62. David M Blei, Thomas L Griffiths, and Michael I Jordan. 2010. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. Journal of the ACM (JACM) 57, 2 (2010), 1–30.
63. Chong Wang and David Blei. 2009. Variational inference for the nested Chinese restaurant process. Advances in Neural Information Processing Systems 22 (2009), 1990–1998.
64. Diederik Kingma and Max Welling. 2014. Efficient gradient-based inference through transformations between bayes nets and neural nets. In Proceedings of ICML.
65. Ponnambalam Kumaraswamy. 1980. A generalized probability density function for double-bounded random processes. Journal of hydrology 46, 1-2 (1980), 79–88.
66. Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In Proceedings of ICRL.
67. Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep Keyphrase Generation. In Proceedings of ACL.
68. Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In Proceeding of CoNLL.
69. Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In Proceedings of ICRL.
70. Jinsong Su, Shan Wu, Deyi Xiong, Yaojie Lu, Xianpei Han, and Biao Zhang. 2018. Variational recurrent neural machine translation. In Proceedings of AAAI.
71. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proceedings of ICLR.
72. Minh N Do. 2003. Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models. IEEE signal processing letters 10, 4 (2003), 115–118.

73. Anette Hulth and Beáta B Megyesi. 2006. A study on automatically extracted keywords in text categorization. In Proceedings of ACL.
74. Mikalai Krapivin, Aliaksandr Autaeu, and Maurizio Marchese. 2009. Large dataset for keyphrases extraction. Technical Report. University of Trento.
75. Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 Task 5 : Automatic Keyphrase Extraction from Scientific Articles. In Proceedings of fifth International Workshop on Semantic Evaluation.
76. Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In Proceedings of ICADL.
77. Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. YAKE! Keyword extraction from single documents using multiple local features. Information Sciences 509 (2020), 257–289.
78. Xiaojun Wan and Jianguo Xiao. 2008. Single Document Keyphrase Extraction Using Neighborhood Knowledge.. In Proceedings of AAAI.
79. Corina Florescu and Cornelia Caragea. 2017. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In Proceedings of ACL.
80. Krutarth Patel and Cornelia Caragea. 2021. Exploiting Position and Contextual Word Embeddings for Keyphrase Extraction from Scientific Papers. In Proceedings of ECACL.
81. Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevillmanning. 1999. KEA: Practical Automatic Keyphrase Extraction. In Proceedings of JCDL
82. Olena Medelyan, Eibe Frank, and Ian H Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In Proceedings of EMNLP.
83. Leo Breiman. 1996. Bagging predictors. Machine learning 24, 2 (1996), 123–140.
84. Yong Zhang, Yang Fang, and Xiao Weidong. 2017. Deep keyphrase generation with a convolutional sequence to sequence model. In Proceedings of ICSAI
85. Wang Chen, Hou Pong Chan, Piji Li, Lidong Bing, and Irwin King. 2019. An Integrated Approach for Keyphrase Generation via Exploring the Power of Retrieval and Extraction. In Proceedings of NAACL.
86. Hou Pong Chan, Wang Chen, Lu Wang, and Irwin King. 2019. Neural Keyphrase Generation via Reinforcement Learning with Adaptive Rewards. In Proceedings of ACL.