Binbin Xie^{*a,c,**}, Jia Song^{*b,c,**}, Liangving Shao^{*a,c,**}, Suhang Wu^{*c*}, Xiangpeng Wei^{*d*}, Baosong Yang^d, Huan Lin^d, Jun Xie^d and Jinsong Su^{a,b,c,**}

^aSchool of Informatics, Xiamen University, Xiamen, 361005, Fujian, China

^bInstitute of Artificial Intelligence, Xiamen University, Xiamen, 361005, Fujian, China

^c Key Laboratory of Digital Protection and Intelligent Processing of Intangible Cultural Heritage of Fujian and Taiwan, Ministry of Culture and Tourism, Xiamen, 361005, Fujian, China

^dAlibaba Group, Hangzhou, 310023, Zhejiang, China

ARTICLE INFO

Keywords: keyphrase prediction automatic keyphrase extraction automatic keyphrase generation deep learning

ABSTRACT

Keyphrase prediction aims to generate phrases (keyphrases) that highly summarizes a given document. Recently, researchers have conducted in-depth studies on this task from various perspectives. In this paper, we comprehensively summarize representative studies from the perspectives of dominant models, datasets and evaluation metrics. Our work analyzes up to 167 previous works, achieving greater coverage of this task than previous surveys. Particularly, we focus highly on deep learning-based keyphrase prediction, which attracts increasing attention of this task in recent years. Afterwards, we conduct several groups of experiments to carefully compare representative models. To the best of our knowledge, our work is the first attempt to compare these models using the identical commonly-used datasets and evaluation metric, facilitating in-depth analyses of their disadvantages and advantages. Finally, we discuss the possible research directions of this task in the future.

1. Introduction

With the rapid development of the Internet and the explosion of information, how to efficiently acquire information from tremendous text data becomes more and more important. To do this, several information compression tasks have been proposed, such as automatic summarization and automatic keyphrase prediction. Compared with other tasks, automatic keyphrase prediction brings forward a higher request to the ability of information compression, since it aims to automatically produce a few keyphrases representing the core contents of the input document. As keyphrases can facilitate understanding documents and provide useful information to downstream tasks, such as information retrieval (Gutwin et al., 1999), document classification (M et al., 2005; Hulth and Megyesi, 2006), document summarization (Zhang et al., 2004; Wang and Cardie, 2013; Pasunuru and Bansal, 2018), question generation (Subramanian et al., 2018) and opinion mining (Wilson et al., 2005; Berend, 2011), automatic keyphrase prediction has attracted increasing attention.

Table 1 shows an example of automatic keyphrase prediction. Generally, keyphrases can be divided into two categories: present keyphrases that continuously appear in the input document and absent keyphrases that do not match any contiguous subsequence of the document. To achieve high-quality keyphrase prediction, early studies mainly focus on automatic keyphrase extraction (Hulth, 2003; Mihalcea and Tarau, 2004; Nguyen and Kan, 2007; Wan and Xiao, 2008), which aims to directly extract keyphrases from the input document. Recently, the rise of deep learning prompts researchers to focus on *automatic keyphrase generation* (Meng et al., 2017; Yuan et al., 2020; Ye et al., 2021b), where

^{*}This work was supported in part by National Natural Science Foundation of China under Grant 62276219, in part by Natural Science Foundation of Fujian Province of China under Grant 2020J06001, and in part by Youth Innovation Fund of Xiamen under Grant 3502Z20206059. (Correspongding author: Jinsong Su.)

^{*}Equally contribution.

^{**} Corresponding authors. Tel.: +86 18750236638

[😫] xdblb@stu.xmu.edu.cn (B. Xie); songjia@stu.xmu.edu.cn (J. Song); liangyingshao@stu.xmu.edu.cn (L. Shao); wush xmu@outlook.com (S. Wu); pemywei@gmail.com (X. Wei); yangbaosong.ybs@alibaba-inc.com (B. Yang); lilai.lh@alibaba-inc.com (H. Lin); gingjing.xj@alibaba-inc.com (J. Xie); jssu@xmu.edu.cn (J. Su)

ORCID(s): 0000-0001-7832-1507 (B. Xie); 0000-0002-0415-7748 (J. Song); 0000-0002-2230-9952 (L. Shao); 0000-0001-5606-7122 (J. Su)

An example of keyphrase prediction and present keyphrases that appear in the document are underlined.

Input Document: A nonmonotonic observation logic. A variant of Reiter's default logic is proposed as a logic for reasoning with <u>defeasible observations</u>. Traditionally, default rules are assumed to represent generic information and the facts are assumed to represent specific information about the situation, but in this paper, the specific information derives from <u>defeasible observations</u> represented by (normal free) default rules, and the facts represent (hard) background knowledge. Whenever the evidence underlying some observation is more refined than the evidence underlying another observation, this is modelled by means of a priority between the default rules representing the observations. We thus arrive at an interpretation of prioritized normal free default logic as an observation logic, and we propose a semantics for this observation logic. Finally, we discuss how the proposed observation logic relates to the multiple extension problem and the problem of sensor fusion.

Keyphrases: defeasible observations; nonmonotonic logic; prioritized default logic

Table 2

Paper publications of keyphrase extraction and keyphrase generation at the main computer science conferences, and '-' denotes that the conference is not held or has not been held yet.

Conf.	2017	2018	2019	2020	2021	2022						
	Keyphrase Extraction											
ACL	2	0	0									
EMNLP	0	0	1	1	3	0						
NAACL	-	1	1	-	2	1						
COLING	-	0	-	4	-	3						
AAAI	3	0	0	0	0	0						
		Keyphra	ise Gene	eration								
ACL	1	0	3	3	2	0						
EMNLP	0	2	0	3	3	5						
NAACL	-	0	2	-	3	2						
COLING	_	0	-	1	-	0						
AAAI	0	0	1	0	1	2						
Total.	6	3	9	11	14	13						

dominant models can generate not only present but also absent keyphrases. Tables 2 shows the number of papers related to automatic keyphrase prediction, published at the main computer science conferences. It can be said that automatic keyphrase prediction has always been one of the research hotpots.

In this paper, we first provide a comprehensive review of automatic keyphrase prediction from the following aspects: dominant models, datasets and evaluation metrics. Compared with previous surveys (Hasan and Ng, 2014; Siddiqi and Sharan, 2015; Çano and Bojar, 2019; Alami Merrouni et al., 2020; Nasar et al., 2019), our work summarizes up to 167 previous works, achieving greater coverage of this task. More importantly, our work is not only the first attempt to thoroughly summarize keyphrase extraction based on neural networks, but also focusing highly on the recent advancements of neural keyphrase generation on different investigated problems. Please note that neural keyphrase generation has become the hot research topic in this community, since it is able to predict not only present keyphrases but also absent keyphrases, which accounts a large proportion in the commonly-used keyphrase generation datasets. Particularly, we further introduce the recent advancements in keyphrase generation, including pre-trained model based keyphrase generation models, echoing with the development trend of natural language processing.

Then, we conduct several groups of experiments to carefully compare representative models, so as to analyze their characteristics. Unlike previous studies generally using different datasets and metrics to evaluate models, we

use the identical commonly-used datasets and evaluation metric to ensure fair comparions among these representative models, and then analyze their advantages and disadvantages in different scenarios. Via our experiments, we can reach some interesting conclusions: 1) Generally, unsupervised extraction models perform worst among all kinds of unsupervised and supervised models. However, when it exists a serious domain discrepancy between the training set and test set, the unsupervised extraction models may achieve comparable performance with the supervised ones. 2) Among three commonly-used paradigms for keyphrase generation, ONE2SET surpasses the others and achieve the best performance, while is still inferior to the extraction models in predicting present keyphrases. 3) Combining with extraction, generation and retrieval-based methods have potential to achieve better overall results for both present and absent keyphrase predictions.

Finally, we point out the future research directions of keyphrase prediction task, which will play a positive role in guiding the follow-up studies. Note that we propose some directions that were not considered in previous surveys, such as multi-modality keyphrase prediction, and multilingual keyphrase prediction.

2. Automatic Keyphrase Extraction

Figure 1 shows the taxonomy of representative studies on automatic keyphrase extraction. This line of research mainly focuses on how to directly extract keyphrases from an input document. Usually, it consists of three steps: 1) applying hand-crafted rules to obtain candidate phrases, such as removing stop words (Liu et al., 2009), applying POS tagging (Mihalcea and Tarau, 2004), extracting n-grams (Witten et al., 1999), and using knowledge bases (Nguyen and Phan, 2009), 2) designing various hand-engineered features to represent candidate keyphrases, and 3) determining the final keyphrases based on features using unsupervised or supervised models.

In the following subsections, we will first briefly introduce the hand-engineered features, and then describe the unsupervised and supervised models using these features in detail.

2.1. Hand-engineered Features

There are mainly four kinds of the internal document-based features used (Witten et al., 1999; Turney, 2002; Hulth, 2003; Zhang et al., 2006; Campos et al., 2018; Ohsawa et al., 1998): statistical features (phrase length, TF-IDF, the number of sentences containing phrases, co-occurrence frequency, etc.), positional features (occurrence positions, sentence boundaries, etc.), linguistics features (POS tags, case information, surrounding words, etc.), and logical structure features (the hierarchy, title, author list of the input document, etc.).

In addition, many features are proposed using the external documents, such as the similarity based on Wikipedia, candidate frequency based on the external documents, citation and web linkage.

2.2. Unsupervised Keyphrase Extraction

Generally, unsupervised models for keyphrase extraction can be roughly divided into statistical models, graph-based models and deep learning-based models, which will be briefly introduced below.

2.2.1. Unsupervised Statistical Models

These models are directly conducted based on the abundant hand-engineered features. Among these features, the most important one is TF-IDF (Salton and Buckley, 1988), which can quantify the importance of each candidate phrase and thus becomes the basis of many follow-up models. For example, El-Beltagy and Rafea (2009) consider the position of each candidate in the input document and introduce a length-related weight to adjust its TF-IDF value. Furthermore, Campos et al. (2018) propose YAKE involving five hand-engineered features: case information, phrase position, term frequency, the frequency of phrase appearing within different sentences, and the number of surrounding words. Based on these features, Won et al. (2019) further determine the number of keyphrases according to the length of the input document.

2.2.2. Unsupervised Graph-based Models

KeyGraph (Ohsawa et al., 1998) is the first graph-based model for keyphrase extraction. In this model, frequently co-occurrent phrases are connected to form a graph, which is then partitioned into subgraphs via clustering. Finally, the importance of each candidate phrase is quantified according to the subgraph based statistical information. Grineva et al. (2009) firstly calculate edge weights as the phrase-level semantic relatedness based on Wikipedia, and then apply the community detection algorithm (Newman and Girvan, 2004) to obtain dense subgraphs, where phrases from the most important subgraphs are considered as keyphrases. Similarly, Liu et al. (2009) construct a word graph and cluster



Figure 1: The Taxonomy of Representative Studies on Automatic Keyphrase Extraction.

words according to the semantic distances based on the word co-occurrence frequency or Wikipedia statistics. Then, the noun phrases expanded from cluster centers are chosen as keyphrases.

Inspired by PageRank (Page et al., 1999), Mihalcea and Tarau (2004) propose TextRank that iteratively conducts importance propagation on a co-occurrent word graph. Along this line, Danesh et al. (2015) extend TextRank by using phrases as graph nodes. Then, many features are explored to adjust edge weights, including statistical features (phrase frequency and length (Danesh et al., 2015), word co-occurrence frequency (Wan and Xiao, 2008)) and position

information (Florescu and Caragea, 2017). Besides, to exploit more contexts, Wan and Xiao (2008), Gollapalli and Caragea (2014) extend the single-document word graph with similar documents and citation network, respectively. In addition to the PageRank-based centrality measure, Vega-Oliveros et al. (2019) consider other commonly-used centrality measures, and then propose an optimal combination of centrality measures to extract keywords from an undirected and unweighted word graph.

Intuitively, ideal keyphrases should be consistent with the topics of the input document. Thus, researchers introduce the topic information to refine graph-based models. Typically, Liu et al. (2010) propose TPR that adopts LDA (Blei et al., 2003) to obtain topic information and then separately performs PageRank for each topic. To alleviate the huge computational cost of TPR, researchers extend TPR into Single Topical PageRank (Sterckx et al., 2015) and SalienceRank (Teneva and Cheng, 2017), both of which perform PageRank once for each document. Compared to the former, the latter can extract not only topic-specific but also corpus-correlated keyphrases. Unlike the above studies based on LDA, Bougouin et al. (2013) propose TopicRank, which firstly clusters similar phrases to form topics and then constructs a topic graph for PageRank. Afterwards, they select the most representative phrases from each topic as keyphrases. To refine TopicRank, Boudin (2018) represents candidate phrases and topics in a single graph and exploits their mutual reinforcement to improve candidate ranking.

2.2.3. Unsupervised Deep Learning-based Models

With the prosperous development of deep learning, researchers introduce neural networks to learn semantic representations of input documents and candidate phrases for ranking, of which studies can be roughly divided into the following four categories: **Phrase-Document Similarity.** The common practice is to measure the importance of each candidate phrase according to the phrase-document representation similarity. To do this, EmbedRank (Bennani-Smires et al., 2018) uses Sent2Vec (Pagliardini et al., 2018) and Doc2Vec (Le and Mikolov, 2014) to represent candidates and input documents as vectors. As an extension, EmbedRank⁺ additionally considers the similarities between candidates to generate diverse keyphrases. Unlike EmbedRank using Sent2vec and Doc2vec, SIFRank (Sun et al., 2020) defines the vector representations of candidates, sentences and input documents as weighted averages of their corresponding ELMo embeddings (Peters et al., 2018), respectively. Further, SIFRank⁺ considers the positions of candidates within the document. Subsequently, Li and Daoutis (2021) improve SIFRank by incorporating domain relevance and phrase quality into ranking scores. Papagiannopoulou and Tsoumakas (2018) use entire documents to learn Glove (Pennington et al., 2014) embeddings, and then rank candidates according to the sum of word-document similarities.

Graph-based Ranking. Besides, researchers apply deep learning to refine the unsupervised models based on phrase graphs. For example, Key2Vec (Mahata et al., 2018) directly trains FastText to learn representations of candidate phrases and document themes, and then uses candidate-theme similarities to adjust the edge weights of PageRank. Similarly, Liang and Zaki (2021) consider the co-occurrence and similarities between candidates for more accurate edge weighting of PageRank. Using embedding-based graph, Asl and Banda (2020) apply PageRank or centrality algorithm to obtain the importance of candidates for ranking. Liang et al. (2021) find that the phrase-document representation similarity (i.e. EmbedRank) is insufficient to capture different contexts for keyphrase extraction. To address this issue, they define a boundary-aware centrality to capture local salient information and positional information of candidates for ranking.

Semantic Importance of Keyphrases. Keyphrases play an important role in the representation learning of the input document. Thus, the representation of the input document will change if any keyphrase is missing. To model this intuition, Zhang et al. (2021) alternatively mask each candidate phrase and evaluate its importance according to the representation difference between the original document and the masked one. Recently, Joshi et al. (2022) adopt a similar strategy that mainly focuses on the change of topic distributions.

Attention Mechanism Information. Different from the above studies based on deep learning similarities, (Ding and Luo, 2021) use self-attention weights to quantify the importance of each candidate phrase within the sentence and measure its semantic relatedness to the document according to its cross-attention weights. Additionally, Gu et al. (2021) generate pseudo keyphrases for unlabeled documents using unsupervised statistic models or an existing knowledge base, and then train a keyphrase classifier fed with the self-attention map from RoBERTa (Zhuang et al., 2021).

2.3. Supervised Keyphrase Extraction

Usually, supervised models for keyphrase extraction can be divided into statistical and deep learning-based models.



Figure 2: The Taxonomy of Representative Studies on Automatic Keyphrase Generation.

2.3.1. Supervised Statistical Models

Similar to unsupervised keyphrase extraction, abundant supervised statistical models leverage well-designed features, including statistical features (Witten et al., 1999; Turney, 2002; Kelleher and Luz, 2005; Haddoud and Abdeddaïm, 2014; Xie et al., 2017), positional features (Frank et al., 1999; Medelyan and Witten, 2006; Zhang, 2008; Jiang et al., 2009), linguistic features (Hulth, 2003; Gollapalli et al., 2017), logical structures (Yih et al., 2006; Zhang et al., 2006; Nguyen and Kan, 2007; Nguyen and Luong, 2010), and external document-based features (Shi et al., 2008; Medelyan et al., 2009; Lopez and Romary, 2010; Gollapalli and Caragea, 2014; Wang and Li, 2017; Zhang et al., 2017a).

Based on these features, researchers model keyphrase extraction as a sequence labeling task (Zhang, 2008; Gollapalli et al., 2017), a binary classification task or a ranking task (Jiang et al., 2009; Zhang et al., 2017a) with various machine learning algorithms, such as conditional random field (Zhang, 2008; Gollapalli et al., 2017), logistic regression (Yih et al., 2006; Shi et al., 2008; Haddoud and Abdeddaïm, 2014), Naive Bayes (Witten et al., 1999; Frank et al., 1999; Kelleher and Luz, 2005; Medelyan and Witten, 2006; Nguyen and Kan, 2007; Nguyen and Luong, 2010; Caragea et al., 2014; Xie et al., 2017), SVM (Zhang et al., 2006), bagged decision trees (Turney, 2002; Medelyan et al., 2009; Lopez and Romary, 2010) and other ensemble models (Hulth, 2003; Wang and Li, 2017).

2.3.2. Supervised Deep Learning-based Models

Wang et al. (2005) first propose a feedforward neural network based classifier for supervised keyphrase extraction. Henceforth, deep learning-based supervised keyphrase extraction has gradually become one of the hot topics.

Sequence Labeling. Supervised keyphrase extraction is often modeled as a deep learning-based sequence labeling task. Typically, Zhang et al. (2016) propose Joint-Layer RNN to extract keyphrases at different discrimination levels: judging whether the current word is a keyword and employing BIOES tagging scheme to identify keyphrases. Based on Joint-Layer RNN, Zhang et al. (2018) introduce conversation context to enrich the vector representations of microblog posts. To simulate the human attention of reading during keyphrase annotating, Zhang and Zhang (2019) integrate an attention mechanism into Joint-Layer RNN. Meanwhile, researchers also explore more features for this model, such as medical concepts from an external knowledge base (Saputra et al., 2018), phonetics, phonological features (Chowdhury et al., 2019), and syntactical features (Mahfuzh et al., 2020).

Also, applying pre-trained models to supervised keyphrase extraction has become dominant. For example, on the basis of SciBERT (Beltagy et al., 2019), (Sahrawat et al., 2019) and (Garg et al., 2020) stack BiLSTM+CRF and LSTM+CRF to identify keyphrases, respectively. Using the same model, (Santosh et al., 2020a) introduce a document-level attention and a gating mechanism to refine representation learning. Wang et al. (2020b) separately leverage BERT and Transformer to encode the document and multi-modal information in web pages for keyphrase extraction. Gero and Ho (2021) use BERT-LSTM or BioBERT-LSTM to obtain the topic representations of input documents, encouraging the extraction of topic-consistent words.

Different from these studies, Santosh et al. (2020b) utilize graph encoders to separately incorporate syntactic and semantic dependency information for better encoder representation. On the basis of the input document and the co-occurence graph, Nikzad-Khasmakhi et al. (2021) adopt BERT and graph embedding techniques to learn the word-level textual and structure representations, which are combined and fed into a sequence labeling tagger.

Binary Classification. Researchers also explore supervised keyphrase extraction as a binary classification task. Xiong et al. (2019) integrate the visual representation of the input document into ELMo word embeddings, and then use a convolutional Transformer to model interactions among candidate phrases for keyphrase classification. Besides, they introduce query prediction as a pre-training task. Prasad and Kan (2019) propose Glocal, an improved GCN, which incorporates the global importance of each node relative to other nodes to learn word representations from a word graph. Based on these representations, keywords are identified via classification and finally used to reconstruct keyphrases via re-ranking.

Ranking. Sarkar et al. (2010) first apply a deep learning-based ranking model to achieve supervised keyphrase extraction. Mu et al. (2020) use BERT stacked with BiLSTM to model semantic interactions among candidate phrases, and then rank them according to the binary classification score and the hinge loss between the considered phrase and others. Sun et al. (2021) propose JointKPE that learns to rank candidate phrases according to their document-level informativeness. Particularly, it is jointly trained with keyphrase chunking to guarantee the phraseness of candidates. Song et al. (2021) investigate three kinds of features for ranking: the syntactic accuracy of the candidate phrase, the information saliency between the candidate and input document, and the concept consistency between the candidate and the input document.

Data Utilization. Based on a word graph, Luan et al. (2017) employ label propagation together with a data selection scheme to leverage unlabeled documents. Lai et al. (2020) propose a self-distillation model for keyphrase extraction. In this approach, a teacher model is trained on labeled examples, while a student model is trained on both labeled examples and pseudo examples generated by the teacher model. During the subsequent training procedure, the teacher model is re-initialized with the student model and repeats the above procedure. To address the issue of incomplete annotated training data, Lei et al. (2021) introduce negative sampling to adjust the training loss on unlabeled data. From a different perspective, Kontoulis et al. (2021) believe that full-texts can provide richer information while containing more noise than the input abstract. Thus, they leverage summaries induced from full-texts to refine keyphrase extraction.

3. Automatic Keyphrase Generation

Unlike the studies on keyphrase extraction, keyphrase generation models can produce absent keyphrases that do not appear in the input document. In this respect, Meng et al. (2017) propose the first keyphrase generation model, CopyRNN, which inspires many subsequent models. Usually, these models are based on an encoder-decoder framework, where the encoder learns the semantic representation of each input document, and then the decoder equipped with a copying mechanism (Gu et al., 2016) automatically produces keyphrases.

In the following subsections, we summarize representative advancements of keyphrase generation according to different investigated problems. The taxonomy of representative studies on automatic keyphrase generation is shown in Figure 2.



Figure 3: The three dominant paradigms for keyphrase generation.

3.1. Paradigms

Generally, paradigms of dominant keyphrase generation models can be classified into ONE2ONE (Meng et al., 2017), ONE2SEQ (Yuan et al., 2020) and ONE2SET (Ye et al., 2021b), as shown in Figure 3.

ONE2ONE. Typically, during model training, each training instance contains an input document and only one corresponding keyphrase from the splitted target keyphrases. During inference, ONE2ONE models adopt beam search to produce candidate phrases and then pick the top-*K* ranked ones as the final keyphrases.

As the earliest paradigm, it has a far-reaching impact but neglects the correlation among keyphrases, limiting the potential of keyphrase generation models.

ONE2SEQ. To deal with the above issue, the ONE2SEQ paradigm models keyphrase generation as a sequence generation task. To this end, target keyphrases are sorted in a predefined order and concated as a sequence with delimiters. Usually, present keyphrases are firstly sorted according to their occurrence, while absent keyphrases are then randomly sorted (Meng et al., 2019, 2021).

Due to the advantage of exploiting the semantic interdependence between keyphrases, ONE2SEQ has become the most commonly-used paradigm. However, its premise of a predefined order introduces a bias into model training,

especially when the order of generated keyphrases is inconsistent with the predefined one. Besides, ONE2SEQ models tend to generate duplicated keyphrases (Chen et al., 2020; Ye et al., 2021b).

ONE2SET. Furthermore, to address the above bias defect of ONE2SEQ, Ye et al. (2021b) propose ONE2SET, where the keyphrase generation is modeled as a set generation task. Typically, its decoder utilizes different learnable control codes to generate a set of keyphrases in parallel. During model training, the training loss is calculated according to the one-to-one alignments between the predicted keyphrases and target ones determined by the Hungarian Algorithm (Kuhn, 1955).

3.2. Document Encoding

Typically, CopyRNN (Meng et al., 2017) adopts RNN as its encoder and thus suffers from low efficiency when handling long documents. To solve this problem, Zhang et al. (2017b) replace RNN with CNN to boost encoding efficiency.

Besides, some researchers argue that sentences should be treated differently due to their unequal importance in document encoding. Chen et al. (2019b) design Title-Guided Network, which additionally uses the title as a query to gather the information of title-relevant words in the input document. Kim et al. (2021b) takes into account useful structures of web documents such as title, body, header, query, to build a word graph representing both position-based proximity and structural relations. Luo et al. (2020) use a selection network to filter unimportant sentences, while Ahmad et al. (2021) apply this network to adjust the weights of the decoder copying mechanism.

Meanwhile, researchers also focus on incorporating more information into the encoder. For instance, Zhao and Zhang (2019) explore linguistic information for document encoding. To alleviate data sparsity in social media, Wang et al. (2019) apply a variational neural network to incorporate topic information into the model.

3.3. Decoding Strategies

Unlike the conventional decoder that can predict both present and absent keyphrases, Sun et al. (2019) propose a diversified Pointer Network decoder for the ONE2ONE paradigm, which only copies a set of diverse present keyphrases.

Meanwhile, more researchers focus on refining the decoding manners under ONE2SEQ paradigm. For example, Chen et al. (2020) propose an exclusive hierarchical decoder that involves two levels of decoding to exploit the phrase-level and word-level correlation for keyphrase generation. Similarly, Santosh et al. (2021b) model the abovementioned hierarchical structure by incorporating a conditional variational autoencoder. Besides, Zhang et al. (2022) propose a hierarchical topic-guided variational neural network by integrating the hierarchical topic information to guide the keyphrases generation. Some researchers argue that uniformly modeling the generation of present and absent keyphrases is unreasonable, since their prediction difficulties are significantly different. Zhao et al. (2021) propose a Select-Guide-Generate decoding strategy, which firstly selects present keyphrases from the input document and then exploits these keyphrases to guide the generation of absent ones. Similarly, Liu et al. (2021) first fine-tune a BERT-based model to identify present keyphrases, to benefit the generation of absent ones. Wu et al. (2021) jointly train present keyphrase extraction and absent keyphrase generation, exploiting their mutual relation via stacker relation layer and bag-of-words constraints. Very recently, Wu et al. (2022b) propose a mask-predict decoder to explore constrained and non-autoregressive generation for absent keyphrase generation.

3.4. Model Training Strategies

Chan et al. (2019) propose a reinforcement learning (RL) approach with an adaptive reward for keyphrase generation. If the model does not generate enough keyphrases, the reward is defined as the recall score that encourages the model to generate enough keyphrases. Otherwise, the F_1 score is used as the reward to prevent the model from over-generating incorrect keyphrases. To ease the synonym problem, Luo et al. (2021) further improve the RL reward function by considering word-level F_1 score, edit distance, duplication rate, and generation quantity.

Besides, researchers apply generative adversarial networks to the keyphrase generation task (Swaminathan et al., 2020a,b; Lancioni et al., 2020), where the generator is trained to produce accurate keyphrases and the discriminator is expected to distinguish machine-generated and human-curated keyphrases.

Many researchers apply multitask model to the keyphrase generation task (Chen et al., 2019a; Ahmad et al., 2021). Typically, Ye and Wang (2018) jointly train keyphrase generation and title generation to improve the generalization ability of the model. Similarly, Zhao and Zhang (2019) introduce POS tagging as an auxiliary task of keyphrase generation.

3.5. Exploitation of External Information

Inspired by the studies of other NLP tasks (Liu et al., 2018; Wang et al., 2018; Zhang et al., 2019), researchers explore the information beyond input documents to generate better keyphrases.

In this regard, Diao et al. (2020) employ a cross-document attention to leverage similar documents for better document encoding. Garg et al. (2021) explore numerous ways to incorporate additional data for keyphrase generation and find that the summary of the article is the most beneficial. Besides, researchers consider the keyphrases of similar documents. Chen et al. (2019a) leverage the retrieved keyphrases from similar documents to guide the keyphrase generation and re-ranking. Santosh et al. (2021a) also collect additional keyphrases from similar documents to automatically form a gazetteer, which is used to enrich the vocabulary for improving keyphrase generation. To exploit both similar documents and their keyphrases, Ye et al. (2021a) construct a heterogeneous keyword-document graph model, which is equipped with a reference-aware decoder to copy words from the input document and its similar ones. To deal with the data without title, Kim et al. (2021a) construct a structure graph using the input document and its related but absent keyphrases retrieved from other documents. This graph can provide structure-aware representations for better keyphrase generation. Besides, Wang et al. (2020c) utilize the rich features embedded in the matching images to explore the joint effects of texts and images for keyphrase prediction.

3.6. Solving Duplication and Coverage Issues of Generated Keyphrases

Chen et al. (2018) point out that the ONE2ONE paradigm neglects the correlation among keyphrases, leading to duplication and coverage issues of generated keyphrases. To solve these issues, they propose CoryRNN that reviews preceding keyphrases to eliminate duplicates, and utilizes the coverage mechanism (Tu et al., 2016) to improve the coverage for keyphrases.

The ONE2SEQ paradigm has the same issues, which become more serious when generating long keyphrase sequences. To deal with this defect, Yuan et al. (2020) employ orthogonal regularization to explicitly distinguish the delimiter-generated hidden states, so as to improve the diversity of generated keyphrases. Bahuleyan and Asri (2020) use an unlikelihood training loss to produce diverse keyphrases. Along this line, Chen et al. (2020) explore not only an training strategy with an exclusive loss, but also an exclusive search strategy to avoid generating duplicate keyphrases. In this way, the model is encouraged to generate keyphrases with different first words.

3.7. Low-resource Keyphrase Generation

The performance of keyphrase generation models deeply depends on the quantity and quality of training data. Unfortunately, the commonly-used labeled datasets are often relatively small, making low-resource keyphrase generation a realistic and valuable research direction

Ye and Wang (2018) propose a semi-supervised model that first generates pseudo keyphrases for unlabeled documents and then use them as incremental training data. Besides, Shen et al. (2022) use unsupervised extraction models to collect keyphrases and then draw pesudo keyphrases for each document based on lexical and semantic level similarities. Finally, the pesudo absent keyphrases are used to train and update the model.

Recently, due to pre-trained models contain abundant knowledge that may benefit keyphrase generation, keyphrase generation based on pre-trained models have received a rising interest. In this respect, Wu et al. (2021) first introduce the pre-trained model UniLM (Dong et al., 2019) into keyphrase generation. Additionally, Garg et al. (2021) utilize Longformer (Beltagy et al., 2020) to deal with the keyphrase generation for long documents. Besides, BART (Lewis et al., 2020), a denoising self-supervised autoencoder, is extensively applied due to its great potential in text generation tasks. For instance, Chowdhury et al. (2022) directly construct an ONE2SEQ model based on the fine-tuned BART. Kulkarni et al. (2022) propose KeyBART, which uses boundary tokens and position embeddings to predict the masked keyphrase and then determine whether a keyphrase is replaced or retained. In addition to the above masked keyphrase prediction, Wu et al. (2022b) apply a prompt-based learning approach for constrained absent keyphrase generation. They firstly define overlapping words between absent keyphrase and document as keywords, and then use a mask-predict decoder to generate the final absent keyphrase under the constraints of prompt.

3.8. Keyphrase Ranking

Due to the property of beam search, ONE2ONE models tend to select short phrases. To deal with this issue, Ni'mah et al. (2019) introduce word-level and ngram-level attention scores to boost the ranking scores of long keyphrases. Besides, Shen et al. (2022) combine the TF-IDF relatedness and embedding-based keyphrase-document cosine similarity

The commonly-used datasets for keyphrase predictions.

Dataset	Domain	Language	Docs
Inspec (Hulth, 2003)	Papers	EN	2.0K
NUS (Nguyen and Kan, 2007)	Papers	EN	211
PubMed (Schutz et al., 2008)	Papers	EN	1.3K
Krapivin (Krapivin et al., 2009)	Papers	EN	2.3K
Citeulike-180 (Medelyan et al., 2009)	Papers	EN	181
SemEval-2010 (Kim et al., 2010)	Papers	EN	244
TALN (Boudin, 2013)	Papers	EN/FR	521/1.2K
KDD (Gollapalli and Caragea, 2014)	Papers	EN	755
WWW (Gollapalli and Caragea, 2014)	Papers	EN	1.3K
TermLTH-Eval (Bougouin et al., 2016)	Papers	FR	400
KP20k (Meng et al., 2017)	Papers	EN	567.8K
LDPK3K (Mahata et al., 2022)	Papers	EN	96.8K
LDPK10K (Mahata et al., 2022)	Papers	EN	1.3M
DUC (Wan and Xiao, 2008)	News	EN	308
110-PT-BN-KP (Marujo et al., 2011)	News	PT	110
500N-KPCrowd (Marujo et al., 2012)	News	EN	500
Wikinews (Bougouin et al., 2013)	News	FR	100
PerKey (Doostmohammadi et al., 2018)	News	PER	553.1K
KPTimes (Gallina et al., 2019)	News	EN	279.9K
Twitter (Zhang et al., 2016)	Tweets	EN	112.5K
Weibo (Wang et al., 2019)	Tweets	ZH	46.3K
Text-Image Tweets (Wang et al., 2020c)	Tweets	EN	53.7K
NZDL (Witten et al., 1999)	Reports	EN	1.8K
Blogs (Grineva et al., 2009)	Web pages	EN	252
StackExchange (Wang et al., 2019)	QA	EN	49.4K

to rank phrases. When reranking phrases, Chen et al. (2019a) also consider phrases retrieved from similar documents and phrases extracted from documents. In addition, Ye and Wang (2018) apply beam search into an ONE2SEQ paradigm based model, which generates multiple candidate phrase sequences and then collect unique keyphrases from the top-ranked beams in descending order.

4. Datasets

The commonly-used datasets for keyphrase prediction are shown in Table 3. According to domains, they could be divided into reports, News, tweets, web pages, QA and scientific articles. Most of these datasets are in English, a few are in French, Persian, Chinese, and Portuguese.

As the most widely-used dataset, KP20k consists of articles in computer science from various online digital libraries. Overall, these datasets are relatively small, which is not applicable to industrial applications. Hence, it is urgent to construct large-quantity and high-quality multilingual datasets, so as to further promote the development of keyphrases prediction.

Considering the tradeoff between cost and quality of expert annotations, Chau et al. (2020) explore multiple annotation strategies, including self review, peer review, and so on.

5. Evaluation Metrics

Let $\hat{Y} = (\hat{y}_1, \hat{y}_2, ..., \hat{y}_m)$ and $Y = (y_2, y_2, ..., y_n)$ to be the predicted and target keyphrases, respectively. The common practice is to use only top *k* predictions with the highest scores for evaluation, where *k* is a pre-defined constant (usually 5 or 10). Particularly, to eliminate the influence of morphology, the predicted keyphrases are stemmed by applying Porter Stemmer¹ (Meng et al., 2017).

The commonly-used metrics include precision, recall and F_1 scores. Early studies use $F_1@5$ and $F_1@10$ to evaluate the quality of generated present keyphrases, and R@5 and R@10 to measure the quality of generated absent keyphrases

¹https://github.com/nltk/nltk/blob/develop/nltk/stem/porter.py

(Meng et al., 2017; Chen et al., 2018, 2019b). Formally, these metrics are defined as follows:

$$P@k = \frac{|\hat{Y}_{:k} \cap Y|}{|\hat{Y}_{:k}|},\tag{1}$$

representing the correct proportion of keyphrases in predictions.

$$R@k = \frac{|\hat{Y}_{:k} \cap Y|}{|Y|},\tag{2}$$

measuring the correct rate of the predicted keyphrase in references.

$$F_1@k = \frac{2P@kR@k}{P@k + R@k},\tag{3}$$

which is a tradeoff between P@k and R@k.

Considering the fact that a model often predicts varying numbers of keyphrases, Yuan et al. (2020) argue that the metrics with the pre-defined constant k cannot accurately evaluate the quality of predicted keyphrases. Thus, they extend $F_1@k$ to two metrics: 1) $F_1@O$: this metric sets k as the number of target keyphrases instead of a pre-defined constant; 2) $F_1@M$: this metric takes all predictions into account. Furthermore, Chan et al. (2019) improve $F_1@M$ by filling target keyphrases with blanks when their number is less than the number of predicted keyphrases.

However, conventional metrics, such as F_1 , which assess the prediction quality at the phrase level, do not take into account the partially matched predictions. To deal with this issue, Luo et al. (2021) propose Fine-Grained (*FG*) evaluation score that considers prediction orders and qualities at the token level, and prediction diversity and numbers at the instance level.

Besides, Habibi and Popescu-Belis (2013) introduce α -*nDCG* (Clarke et al., 2008) to measure the diversity of predicted keyphrases, where *nDCG* represents Normalized Discounted Cumulative Gain measure (Järvelin and Kekäläinen, 2002) and the parameter α is a trade-off between relevance and diversity. Chan et al. (2019) measure the mean absolute error (*MAE*) between the number of predicted keyphrases and the number of target keyphrases. In addition, Chen et al. (2020) define *DupRatio* to evaluate the duplication rate of the predicted keyphrases.

5.1. Implementation Details

In the experiments of keyphrase extraction, we consider the following typical unsupervised models: statistical models including TF-IDF (Salton and Buckley, 1988), YAKE (Campos et al., 2018), graph-based models consisting of TextRank (Mihalcea and Tarau, 2004), SingleRank (Wan and Xiao, 2008), TopicRank (Bougouin et al., 2013), PositionRank (Florescu and Caragea, 2017), MultipartieRank (Boudin, 2018), and deep learning-based models such as EmbedRank (Bennani-Smires et al., 2018), SIFRank (Sun et al., 2020), SIFRank+ (Sun et al., 2020), UKERank (Liang et al., 2021), and JointKPE (Sun et al., 2021). Please Note that JointKPE is the current SOTA supervised model. In the experiments of keyphrase generation, we compare the typical generation models under three kinds of common paradigms: 1) ONE2ONE, CopyRNN (Meng et al., 2017) and KG-KE-KR-M (Chen et al., 2019a), 2) ONE2SEQ, CatSeq (Yuan et al., 2020), catSeqTG-2RF₁ (Chan et al., 2019) and Transformer (Ye et al., 2021b), and 3) ONE2SET, SetTrans (Ye et al., 2021b) and WR-SetTrans(Xie et al., 2022). Besides, we compare the performance of large language models in keyphrase prediction, including BART (Lewis et al., 2020), T5 (Raffel et al., 2020), KeyBART (Kulkarni et al., 2022) and ChatGPT².

During model training, we strictly use the same experiment settings as their original papers. For the model involving multiple variants, we only report the performance of its variant with the best performance. Particularly, following Yuan et al. (2020), we use two experimental settings for ONE2SEQ paradigm models. When using ChatGPT, we explore three commonly-used settings, including zero-shot³, 1-shot, and 5-shot. Specifically, we retrieve the most relevant training instances for the given input document according to the cosine distance of the MiniLM (Wang et al., 2020a) embedding. These pertinent training instances are concatenated at the beginning of the input document and then fed into the ChatGPT to obtain the ultimate predictions for keyphrases. Particularly, to alleviate the instability of neural networks, we run the generation models for 3 times with different seeds and report the average results. Finally, we evaluate the present keyphrase and absent keyphrase predictions, respectively.

²https://chat.openai.com/chat

 $^{^{3}}$ We use the official released prompt (https://platform.openai.com/examples/default-keywords) for keyphrase prediction.

Results of present keyphrase prediction using extraction models. To ensure fair comparsions, we only use the target present keyphrase to evaluate the performance of extraction models, while the previous studies use all target keyphrases.

Model	Ins F1@5	pec F1@M	NL F1@5	JS F1@M	Krap F1@5	ivin F1@M	Sem F1@51	Eval F1@M	KP2 F1@51	2 0k =1@M		
Unsuper	Unsupervised Statistical Extraction Models											
TF-IDF (Salton and Buckley, 1988) YAKE (Campos et al., 2018)	0.132 0.183	0.175 0.193	0.214 0.221	0.213 0.212	0.145 0.188	0.131 0.131	0.151 0.202	0.190 0.204	0.172 0.189	0.146 0.145		
Unsupervised Graph-based Extraction Models												
TextRank (Mihalcea and Tarau, 2004) SingleRank (Wan and Xiao, 2008) TopicRank (Bougouin et al., 2013) PositionRank (Florescu and Caragea, 2017) MultipartiteRank (Boudin, 2018)	0.321 0.325 0.266 0.306 0.269	0.363 0.362 0.301 0.338 0.322	0.092 0.151 0.210 0.228 0.244	0.169 0.195 0.154 0.208 0.188	0.118 0.152 0.168 0.186 0.181	0.144 0.147 0.118 0.143 0.132	0.093 0.146 0.201 0.245 0.227	0.200 0.212 0.163 0.229 0.206	0.091 0.134 0.167 0.183 0.185	0.120 0.131 0.114 0.138 0.132		
Unsupervised	Deep L	earning	g-based	d Extra	ction M	odels						
EmbedRank (Bennani-Smires et al., 2018) SIFRank (Sun et al., 2020) SIFRank+ (Sun et al., 2020) UKERank (Liang et al., 2021)	0.333 0.368 0.348 0.350	0.376 0.385 0.384 0.384	0.166 0.143 0.246 0.238	0.199 0.193 0.203 0.202	0.167 0.164 0.194 0.187	0.150 0.151 0.153 0.162	0.185 0.165 0.244 0.250	0.233 0.213 0.223 0.228	0.153 0.138 0.195 0.178	0.135 0.133 0.138 0.138		
Supervised D	Supervised Deep Learning-based Extraction Models											
Sequence Tagging(Roberta-base) (Sun et al., 2021) JointKPE (Sun et al., 2021)	0.331	0.336	0.321	0.177 0.335	0.476	0.319	0.379 0.393	0.291 0.306	0.416 0.417	0.240 0.239		

Table 5

Results of present keyphrase prediction using generation models. #bs denotes beam size. † indicates previously reported scores.

Model	Insj	bec	Nl	JS	Kra p	pivin	Sem	Eval	KP 2	20k
	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
ONE2ONE Paradigm-based Models										
CopyRNN(#bs=200) (Meng et al., 2017)	0.272	0.293	0.356	0.306	0.283	0.214	0.294	0.257	0.336	0.255
KG-KE-KR-M(#bs=200) (Chen et al., 2019a)	0.324	0.362	0.421	0.342	0.304	0.273	0.325	0.293	0.400	0.277
ONE2SEQ Paradigm-based Models										
CatSeq($\#bs=1$) (Yuan et al., 2020)	0.229	0.266	0.324	0.394	0.270	0.344	0.245	0.296	0.292	0.365
CatSeq($\#bs=50$) (Yuan et al., 2020)	0.328	0.398	0.417	0.395	0.352	0.316	0.343	0.334	0.360	0.302
catSeqTG-2RF ₁ ($\#bs=1$) (Chan et al., 2019)	0.253	0.301	0.375	0.433	0.300	0.369	0.287	0.329	0.321	0.386
Transformer($\#bs=1$) (Ye et al., 2021b)	0.285	0.331	0.371	0.418	0.308	0.356	0.287	0.319	0.330	0.373
ONE2SET Paradigm-based Models										
SetTrans(# <i>bs</i> =1) (Ye et al., 2021b)	0.281	0.318	0.406	0.452	0.339	0.374	0.322	0.354	0.354	0.390
WR-SetTrans(# <i>bs</i> =1) (Xie et al., 2022)	0.330	0.351	0.428	0.452	0.360	0.362	0.360	0.370	0.370	0.378

6. Comparison between Existing Models

To better understand advantages and disadvantages of different models, we conduct several groups of experiments to compare representative models in different settings. To this end, we use the KP20k training set to train various

Results of absent keyphrase prediction using generation models.

Model		pec	NI	JS	Krap	Divin	Sem	Eval	KP2	20k	
	ri@5	FIWM	FI@5		FI@5	FIWM	F1@5	FIWN	FI@5	FI@M	
ONE2ONE Paradigm-based Models											
CopyRNN(#bs=200) (Meng et al., 2017)	0.007	0.007	0.009	0.012	0.013	0.019	0.007	0.011	0.011	0.013	
KG-KE-KR-M(#bs=200) (Chen et al., 2019a) [†]	0.024	0.028	0.060	0.076	0.059	0.063	0.031	0.040	0.070	0.083	
ONE2SEQ Paradigm-based Models											
CatSeq(#bs=1) (Yuan et al., 2020)	0.005	0.009	0.015	0.026	0.018	0.034	0.015	0.022	0.014	0.030	
CatSeq(#b=50) (Yuan et al., 2020)	0.021	0.028	0.038	0.052	0.051	0.065	0.030	0.038	0.041	0.058	
catSeqTG-2RF ₁ (#bs=1) (Chan et al., 2019)	0.012	0.021	0.019	0.031	0.030	0.053	0.021	0.030	0.027	0.050	
Transformer(#bs=1) (Ye et al., 2021b)	0.008	0.017	0.028	0.050	0.030	0.055	0.016	0.022	0.021	0.043	
One	ONE2SET Paradigm-based Models										
SetTrans(#bs=1) (Ye et al., 2021b)	0.018	0.029	0.041	0.061	0.046	0.073	0.029	0.035	0.035	0.056	
WR-SetTrans(#bs=1) (Xie et al., 2022)	0.025	0.034	0.057	0.071	0.057	0.074	0.040	0.043	0.050	0.064	



Figure 4: The training losses of representative models under three paradigms.

models, and then apply the same script⁴ to evaluate the model predictions on five commonly-used test sets: Inspec, NUS, Krapivin, SemEval, and KP20k.

6.1. Comparison of Extraction Models

The performance of extraction models is reported in Table 4. Note that the previous studies in this aspect report the evaluation scores with respect to all target keyphrases. To ensure fair comparisons, we only use the present keyphrases to evaluate the performance of various extraction models.

Overall, unsupervised statistical extraction models perform worst in this setting, and unsupervised graph-based extraction models surpass statistical ones. This result is not surprising, because unsupervised graph-based extraction models not only use statistical features but also employ effective graph algorithms, such as clustering, graph propagation, etc. Moreover, due to the advantage of semantic representation learning, deep learning-based models achieve the best result, echoing the development trend of natural language processing studies from statistical models to deep learning-based models.

Besides, comparing unsupervised and supervised extraction models, we can observe that supervised extraction models outperform unsupervised ones on most test sets except Inspec. Further analysis on Inspec will be provided in

⁴https://github.com/kenchan0226/keyphrase-generation-rl/blob/master/evaluate_prediction.py

Dataset	#pre. KP/doc	#abs. KP/doc	#token/doc	Length of pre. KP	Length of abs. KP	#doc
Inspec	7.23	2.59	134.10	2.44	2.72	500
NUS	6.34	5.31	230.13	1.95	2.56	211
Krapivin	3.26	2.59	189.32	2.16	2.29	400
SemEval	6.25	8.41	245.89	2.08	2.61	100
KP20k	3.24	2.84	179.02	1.85	2.55	570, 802

Table /				
Statistical	features c	of five	datasets.	

Section 6.2.

6.2. Comparison of Gneration Models

Three Training Paradigms Figure 4 shows the training losses of CopyRNN, CatSeq and SetTrans, which are the representative models under three paradigms. CopyRNN suffers from the highest loss, due to the difficulty of model training brought by the One2One paradigm where one input corresponds to multiple targets. One2Seq paradigm alleviates the problem of inconsistent training instances by concatenating target keyphrases into a sequence and Cat2Seq achieves a relatively lower loss than CopyRNN. Among three representative models, SetTrans has the lowest loss after convergence, demonstrating the advantage of the ONE2SET paradigm.



Figure 5: The first occurrence position distribution of present keyphrases in input documents

Comparison of SOTA Extraction Model and Generation Models From the last rows of Table 4 and Table 5, we observe that JointKPE (Sun et al., 2021) outperfoms all generation models in terms of $F_1@5$. However, extraction models cannot dynamically decide the number of extracted keyphrases. If the pre-defined number of extracted keyphrases is larger than the actual number of target keyphrases, it may introduce noise into the extracted phrases, resulting in a low $F_1@M$. Worse still, extraction models are unable to deal with the predictions of absent keyphrases, which account for a large proportion of target keyphrases. Therefore, we argue that a combination of extraction and generation model, such as KG-KE-KR-M (Chen et al., 2019a), has the potential to achieve better overall results than single-mode models.

Back to Table 5, KG-KE-KR-M performs significantly better than CopyRNN, proving the superiority of combing generation and extraction. Note that although KG-KE-KR-M incorporates retrieval and reranking techniques into ONE2ONE paradigm, SetTrans(Ye et al., 2021b) still outperforms KG-KE-KR-M and other generation models in F1@M without special techniques, showing its advantages in predicting the keyphrase number for documents.



Figure 6: The first occurrence position distributions of the target present keyphrases and present keyphrases predicted by SIFRank, JointKPE and SetTrans in Inspec. The document has been divided into ten equal parts, and the x-axis indicates the index of the divided sub-document, for example, x = p1 means that the first 10% of the document. The y-axis is the proportion of the keyphrase in this sub-document.

Analysis of the Inspec Dataset From Table 4 and Table 5, we observe that unsupervised deep learning-based extraction models achieve comparable or better performance than supervised deep learning-based extraction and generation models when predicting present keyphrases on Inspec. To explain this phenomena, we further conduct the following analyses:

1) Table 7 shows the statistical features of datasets. Compared with other test sets, Inspec has the shortest average document length, the longest average length of present keyphrase, and the maximum number of present keyphrase, indicating Inspec is more suitable for extraction models than other datasets.

2) In Figure 5, we also visualize the occurrence position distributions of present keyphrases in each dataset. It reveals that the present keyphrases of the KP20k training set tend to occur in the front of the document. This phenomenon becomes even more evident when analyzing the first occurrence positions of present keyphases. As a result, the supervised models trained on KP20k tend to predict present keyphrases from the front of the document, which, however, is not applicable for Inspec, of which present keyphrases distribute evenly in the document.

3) Figure 6 depicts the position distributions of the target present keyphrases and present keyphrases predicted by SIFRank, JointKPE, and SetTrans in Inspec. Please note that they are the best unsupervised keyphrase extraction, supervised keyphrase extraction and keyphrase generation models, respectively. The distribution of present keyphrases predicted by SIFRank is very close to the distribution of Inspec, while other supervised models are quite different. It supports our hypothesis that supervised models, are deeply affected by the occurrence position distribution of keyphrases in training data, which leads to the degradation of model performance when the test set is domain-mismatch with the training data.

6.3. Comparison of LLMs

Recently, large language models (LLMs) have achieved remarkable success in various NLP tasks and have displayed a variety of capabilities. To evaluate the keyphrase prediction ability of these models, we compare the performance of commonly-used LLMs, including BART, T5, KeyBART, and ChatGPT, across five benchmark datasets.

Table 8 and Table 9 reports the experimental results. We find that compared with previous SOTA models, such as CatSeq, Transformer, SetTrans and WR-SetTrans, LLMs show modest improvements in both present and absent keyphrase predictions. Additionally, our comparison of SetTrans and LLMs suggests that the impact of increasing model parameters is overshadowed by the adoption of new training and inference paradigms.

By synthesizing all results of Table 4, Table 8 and Table 9, we conclude that ChatGPT outperforms all other unsupervised keyphrase extraction methods in terms of F1@5-score and F1@M-score under the zero-shot setting, but is still inferior to the existing SOTA supervised models on almost all datasets. With more training instances, the prediction ability of ChatGPT for both present and absent keyphrases can be significantly improved. Its superior performance on

Results of present keyphrase prediction using large language models. *#bs* denotes beam size. [†] indicates previously reported scores.

	Inspec		NUS		Krapivin		SemEval		KP20k	
Model	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@51	F1@M
CatSeq(#bs=1) (Yuan et al., 2020)	0.229	0.266	0.324	0.394	0.270	0.344	0.245	0.296	0.292	0.365
Transformer(#bs=1) (Ye et al., 2021b)	0.285	0.331	0.371	0.418	0.308	0.356	0.287	0.319	0.330	0.373
SetTrans(#bs=1) (Ye et al., 2021b)	0.281	0.318	0.406	0.452	0.339	0.374	0.322	0.354	0.354	0.390
WR-SetTrans(#bs=1) (Xie et al., 2022)	0.330	0.351	0.428	0.452	0.360	0.362	0.360	0.370	0.370	0.378
BART-base(#bs=1) (Lewis et al., 2020)	0.270	0.323	0.366	0.424	0.270	0.336	0.271	0.321	0.322	0.388
BART-large(#bs=1) (Lewis et al., 2020)	0.276	0.333	0.380	0.435	0.284	0.347	0.274	0.311	0.332	0.392
T5-base(#bs=1) (Raffel et al., 2020)	0.288	0.339	0.388	0.440	0.302	0.350	0.295	0.326	0.336	0.388
T5-large(#bs=1) (Raffel et al., 2020)	0.295	0.343	0.398	0.438	0.315	0.359	0.297	0.321	0.343	0.393
KeyBART(#bs=1) (Kulkarni et al., 2022)	0.268	0.325	0.373	0.430	0.287	0.365	0.260	0.289	0.325	0.398
zero-shot ChatGPT(#bs=1)	0.309	0.428	0.338	0.258	0.237	0.189	0.274	0.252	0.192	0.158
1-shot ChatGPT(#bs=1)	0.421	0.480	0.355	0.359	0.297	0.298	0.319	0.326	0.298	0.295
5-shot ChatGPT(#bs=1)	0.431	0.497	0.365	0.351	0.285	0.287	0.312	0.300	0.297	0.288

Table 9

Results of absent keyphrase prediction using large language models.

	Inspec		NUS		Krapivin		SemEval		KP20k	
Model	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
CatSeq(#bs=1) (Yuan et al., 2020)	0.005	0.009	0.015	0.026	0.018	0.034	0.015	0.022	0.014	0.030
Transformer(#bs=1) (Ye et al., 2021b)	0.008	0.017	0.028	0.050	0.030	0.055	0.016	0.022	0.021	0.043
SetTrans(#bs=1) (Ye et al., 2021b)	0.018	0.029	0.041	0.061	0.046	0.073	0.029	0.035	0.035	0.056
WR-SetTrans(#bs=1) (Xie et al., 2022)	0.025	0.034	0.057	0.071	0.057	0.074	0.040	0.043	0.050	0.064
BART-base(#bs=1) (Lewis et al., 2020)	0.010	0.017	0.026	0.042	0.028	0.049	0.016	0.021	0.022	0.042
BART-large(#bs=1) (Lewis et al., 2020)	0.015	0.024	0.031	0.048	0.031	0.051	0.019	0.024	0.027	0.047
T5-base(#bs=1) (Raffel et al., 2020)	0.011	0.020	0.027	0.051	0.023	0.043	0.014	0.020	0.017	0.034
T5-large(#bs=1) (Raffel et al., 2020)	0.011	0.021	0.025	0.042	0.023	0.045	0.015	0.020	0.017	0.035
KeyBART(#bs=1) (Kulkarni et al., 2022)	0.014	0.023	0.031	0.055	0.036	0.064	0.016	0.022	0.026	0.047
zero-shot ChatGPT(#bs=1)	0.014	0.027	0.003	0.005	0.002	0.004	0.002	0.003	0.025	0.030
1-shot ChatGPT(#bs=1)	0.027	0.048	0.011	0.017	0.015	0.028	0.009	0.011	0.015	0.027
5-shot ChatGPT(#bs=1)	0.028	0.046	0.010	0.015	0.016	0.031	0.016	0.021	0.015	0.027

multiple benchmark datasets highlights its significance for practical applications in various domains. Further research could explore the more effective use of ChatGPT to fully exert its potential.

7. Future Directions

In summary, automatic keyphrase prediction has attracted extensive attention from academia and industry currently. However, it still remains a challenging task in the following aspects:

1) The quality of generated absent keyphrases directly determines the availability of keyphrase generation models. However, dominant models are still unable to produce satisfactory absent keyphrases. Therefore, how to improve the prediction performance on absent keyphrases will be the focus of future research.

2) Intuitively, humans often exploit the information beyond the input document to predict keyphrases. Hence, how to fully exploit more information, such as the extra information from external knowledge base or pre-trained model, for better keyphrase predictions is worth exploring.

3) Short videos have recently emerged as a widespread type of social media due to the explosive growth of the Internet. Two new forms of multi-modal information introduced in the search and recommendation scenarios, video

and audio, place additional demand on keyphrase prediction. Thus, we believe that multi-modal keyphrase prediction is also the future development trend of keyphrase prediction.

4) Existing studies mainly focus on using domain-specific data to train models, such as scientific documents. However, it is unable to handle different domains of data from the Internet. Consequently, how to effectively transfer these models to other domains becomes one problem to be solved in practical applications.

5) The conventional evaluation metrics mainly focus on the comparison between the surface representations of stemmed phrases. However, two phrases may possess the same meaning although their expressions are different. Hence, the quality evaluation of generated keyphrases should consider the comparison between semantic representations of phrases and the application effect in downstream tasks such as retrieval systems (Boudin et al., 2020).

6) Dominant studies model the generations of present and absent keyphrases in a unified manner, although their prediction difficulties vary greatly. Intuitively, it is more reasonable to individually model the generations of absent and present keyphrases. Please note that Wu et al. (2022b) verifies the feasibility of this direction.

7) The generation of keyphrases can draw lesson from the process of human reading and refining keyphrases. For example, humans tend to distill the overall idea first and grasp the specifics later, and thus, an ideal process for keyphrase prediction is to predict keyphrase in a coarse-to-fine manner.

8) Very recently, ChatGPT has demonstrated effectiveness proficiency across a range of NLP tasks. As such, it is imperative to explore the optimal utilization of ChatGPT in keyphrase prediction, in order to fully exert its remarkable potential.

References

- Ahmad, W., Bai, X., Lee, S., Chang, K.W., 2021. Select, extract and generate: Neural keyphrase generation with layer-wise coverage attention, in: Proc. ACL-IJCNLP Conf., pp. 1389–1404. doi:10.18653/v1/2021.acl-long.111.
- Alami Merrouni, Z., Frikh, B., Ouhbi, B., 2020. Automatic keyphrase extraction: a survey and trends. J. Intell. Inf. Syst., 391–424doi:10.1007/ s10844-019-00558-9.
- Asl, J.R., Banda, J.M., 2020. Gleake: Global and local embedding automatic keyphrase extraction. CoRR doi:10.48550/arXiv.2005.09740, arXiv:2005.09740.
- Bahuleyan, H., Asri, L.E., 2020. Diverse keyphrase generation with neural unlikelihood training, in: Proc. COLING Conf., pp. 5271–5287. doi:10.18653/v1/2020.coling-main.462.
- Beltagy, I., Lo, K., Cohan, A., 2019. Scibert: A pretrained language model for scientific text, in: Proc. EMNLP-IJCNLP Conf., pp. 3613–3618. doi:10.48550/arXiv.1903.10676.
- Beltagy, I., Peters, M.E., Cohan, A., 2020. Longformer: The long-document transformer. CoRR doi:10.48550/ARXIV.2004.05150.
- Bennani-Smires, K., Musat, C., Jaggi, M., Hossmann, A., Baeriswyl, M., 2018. Embedrank: Unsupervised keyphrase extraction using sentence embeddings. CoRR doi:10.48550/ARXIV.1801.04470, arXiv:1801.04470.
- Berend, G., 2011. Opinion expression mining by exploiting keyphrase extraction, in: Proc. IJCNLP Conf., pp. 1162–1170.
- Blei, D.M., Ng, A.Y., Jordan, M.I., 2003. Latent dirichlet allocation. J. Mach. Learn. Res., 993–1022.
- Boudin, F., 2013. TALN archives : a digital archive of french research articles in natural language processing, in: Proc. TALN Conf., pp. 507-514.
- Boudin, F., 2018. Unsupervised keyphrase extraction with multipartite graphs, in: Proc. NAACL, Short Papers Conf., pp. 667–672. doi:10.18653/ v1/n18-2105.
- Boudin, F., Gallina, Y., Aizawa, A., 2020. Keyphrase generation for scientific document retrieval, in: Proc. ACL Conf., p. 00. doi:10.18653/v1/2020.acl-main.105.
- Bougouin, A., Barreaux, S., Romary, L., Boudin, F., Daille, B., 2016. TermITH-eval: a French standard-based resource for keyphrase extraction evaluation, in: Proc. LREC Conf., pp. 1924–1927.
- Bougouin, A., Boudin, F., Daille, B., 2013. Topicrank: Graph-based topic ranking for keyphrase extraction, in: Proc. IJCNLP Conf., pp. 543-551.
- Campos, R., Mangaravite, V., Pasquali, A., Jorge, A.M., Nunes, C., Jatowt, A., 2018. Yake! collection-independent automatic keyword extractor, in: Proc. ECIR Conf., pp. 806–810. doi:10.1007/978-3-319-76941-7_80.
- Çano, E., Bojar, O., 2019. Keyphrase generation: A multi-aspect survey, in: Proc. FRUCT Conf., pp. 85–94. doi:10.23919/FRUCT48121. 2019.8981519.
- Caragea, C., Bulgarov, F.A., Godea, A., Das Gollapalli, S., 2014. Citation-enhanced keyphrase extraction from research papers: A supervised approach, in: Proc. EMNLP Conf., pp. 1435–1446. doi:10.3115/v1/d14-1150.
- Chan, H.P., Chen, W., Wang, L., King, I., 2019. Neural keyphrase generation via reinforcement learning with adaptive rewards, in: Proc. ACL Conf., p. 00. doi:10.18653/v1/p19-1208.
- Chau, H., Balaneshin, S., Liu, K., Linda, O., 2020. Understanding the tradeoff between cost and quality of expert annotations for keyphrase extraction, in: Proc. LAW@COLING Conf., pp. 74–86.
- Chen, J., Zhang, X., Wu, Y., Yan, Z., Li, Z., 2018. Keyphrase generation with correlation constraints, in: Proc. EMNLP Conf., pp. 4057–4066. doi:10.18653/v1/d18-1439.
- Chen, W., Chan, H.P., Li, P., Bing, L., King, I., 2019a. An integrated approach for keyphrase generation via exploring the power of retrieval and extraction, in: Proc. NAACL Conf., pp. 2846–2856. doi:10.48550/arXiv.1904.03454.
- Chen, W., Chan, H.P., Li, P., King, I., 2020. Exclusive hierarchical decoding for deep keyphrase generation, in: Proc. ACL Conf., pp. 1095–1105. doi:10.18653/v1/2020.acl-main.103.

- Chen, W., Gao, Y., Zhang, J., King, I., Lyu, M.R., 2019b. Title-guided encoding for keyphrase generation, in: Proc. AAAI Conf., pp. 6268–6275. doi:10.1609/aaai.v33i01.33016268.
- Chowdhury, J.R., Caragea, C., Caragea, D., 2019. Keyphrase extraction from disaster-related tweets, in: Proc. WWW Conf., pp. 1555–1566. doi:10.1145/3308558.3313696.
- Chowdhury, M.F.M., Rossiello, G., Glass, M.R., Mihindukulasooriya, N., Gliozzo, A., 2022. Applying a generic sequence-to-sequence model for simple and effective keyphrase generation. CoRR doi:10.48550/ARXIV.2201.05302, arXiv:2201.05302.
- Clarke, C.L., Kolla, M., Cormack, G.V., Vechtomova, O., Ashkan, A., Büttcher, S., MacKinnon, I., 2008. Novelty and diversity in information retrieval evaluation, in: Proc. SIGIR Conf., pp. 659–666. doi:10.1145/1390334.1390446.
- Danesh, S., Sumner, T., Martin, J.H., 2015. SGRank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction, in: Proc. *SEM@NAACL Conf., pp. 117–126. doi:10.18653/v1/s15-1013.
- Diao, S., Song, Y., Zhang, T., 2020. Keyphrase generation with cross-document attention. CoRR doi:10.48550/arXiv.2004.09800.
- Ding, H., Luo, X., 2021. AttentionRank: Unsupervised keyphrase extraction using self and cross attentions, in: Proc. EMNLP Conf., pp. 1919–1928. doi:10.18653/v1/2021.emnlp-main.146.
- Dong, L., Yang, N., Wang, W., Wei, F., Liu, X., Wang, Y., Gao, J., Zhou, M., Hon, H., 2019. Unified language model pre-training for natural language understanding and generation, in: Proc. NeurIPS Conf., pp. 13042–13054.
- Doostmohammadi, E., Bokaei, M.H., Sameti, H., 2018. Perkey: A persian news corpus for keyphrase extraction and generation, in: Proc. IST Conf., pp. 460–465. doi:10.1109/ISTEL.2018.8661095.
- El-Beltagy, S.R., Rafea, A.A., 2009. Kp-miner: A keyphrase extraction system for english and arabic documents. Inf. Syst., 132–144doi:10.1016/j.is.2008.05.002.
- Florescu, C., Caragea, C., 2017. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents, in: Proc. ACL Conf., pp. 1105–1115. doi:10.18653/v1/P17-1102.
- Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G., 1999. Domain-specific keyphrase extraction, in: Proc. IJCAI Conf., pp. 668–673.
- Gallina, Y., Boudin, F., Daille, B., 2019. Kptimes: A large-scale dataset for keyphrase generation on news documents, in: Proc. INLG Conf., pp. 130–135. doi:10.18653/v1/W19-8617.
- Garg, A., Kagi, S.S., Singh, M., 2020. SEAL: scientific keyphrase extraction and classification, in: Proc. JCDL Conf., pp. 527–528. doi:10.1145/ 3383583.3398625.
- Garg, K., Chowdhury, J.R., Caragea, C., 2021. Keyphrase generation beyond the boundaries of title and abstract. CoRR doi:10.48550/ARXIV. 2112.06776, arXiv:2112.06776.
- Gero, Z., Ho, J.C., 2021. Word centrality constrained representation for keyphrase extraction, in: Proc. BioNLP@NAACL Conf., pp. 155–161. doi:10.18653/v1/2021.bionlp-1.17.
- Gollapalli, S.D., Caragea, C., 2014. Extracting keyphrases from research papers using citation networks, in: Proc. AAAI Conf., pp. 1629–1635. doi:10.1609/aaai.v28i1.8946.
- Gollapalli, S.D., Li, X., Yang, P., 2017. Incorporating expert knowledge into keyphrase extraction, in: Proc. AAAI Conf., pp. 3180–3187. doi:10.1609/aaai.v31i1.10986.
- Grineva, M., Grinev, M., Lizorkin, D., 2009. Extracting key terms from noisy and multitheme documents, in: Proc. WWW Conf., pp. 661–670. doi:10.1145/1526709.1526798.
- Gu, J., Lu, Z., Li, H., Li, V.O., 2016. Incorporating copying mechanism in sequence-to-sequence learning, in: Proc. ACL Conf., pp. 1631–1640. doi:10.18653/v1/P16-1154.
- Gu, X., Wang, Z., Bi, Z., Meng, Y., Liu, L., Han, J., Shang, J., 2021. Ucphrase: Unsupervised context-aware quality phrase tagging, in: Proc. KDD Conf., pp. 478–486. doi:10.1145/3447548.3467397.
- Gutwin, C., Paynter, G.W., Witten, I.H., Nevill-Manning, C.G., Frank, E., 1999. Improving browsing in digital libraries with keyphrase indexes. Decis. Support Syst., 81–104doi:10.1016/S0167-9236(99)00038-X.
- Habibi, M., Popescu-Belis, A., 2013. Diverse keyword extraction from conversations, in: Proc. ACL, Short Papers Conf., pp. 651-657.
- Haddoud, M., Abdeddaïm, S., 2014. Accurate keyphrase extraction by discriminating overlapping phrases. J. Inf. Sci., 488–500doi:10.1177/ 0165551514530210.
- Hasan, K.S., Ng, V., 2014. Automatic keyphrase extraction: A survey of the state of the art, in: Proc. ACL Conf., pp. 1262–1273. doi:10.3115/ v1/p14-1119.
- Hulth, A., 2003. Improved automatic keyword extraction given more linguistic knowledge, in: Proc. EMNLP Conf., pp. 216–223. doi:10.3115/1119355.1119383.
- Hulth, A., Megyesi, B., 2006. A study on automatically extracted keywords in text categorization, in: Proc. ACL Conf., pp. 537–544. doi:10. 3115/1220175.1220243.
- Järvelin, K., Kekäläinen, J., 2002. Cumulated gain-based evaluation of IR techniques. ACM Trans. Inf. Syst., 422–446doi:10.1145/582415. 582418.
- Jiang, X., Hu, Y., Li, H., 2009. A ranking approach to keyphrase extraction, in: Proc. SIGIR Conf., pp. 756–757. doi:10.1145/1571941. 1572113.
- Joshi, R., Balachandran, V., Saldanha, E., Glenski, M., Volkova, S., Tsvetkov, Y., 2022. Unsupervised keyphrase extraction via interpretable neural networks. CoRR doi:10.48550/ARXIV.2203.07640.
- Kelleher, D., Luz, S., 2005. Automatic hypertext keyphrase detection, in: Proc. IJCAI Conf., pp. 1608–1609.
- Kim, J., Jeong, M., Choi, S., Hwang, S., 2021a. Structure-augmented keyphrase generation, in: Proc. EMNLP Conf., pp. 2657–2667. doi:10.18653/v1/2021.emnlp-main.209.
- Kim, J., Song, Y., Hwang, S., 2021b. Web document encoding for structure-aware keyphrase extraction, in: Proc. SIGIR Conf., pp. 1823–1827. doi:10.1145/3404835.3463067.

- Kim, S.N., Medelyan, O., Kan, M., Baldwin, T., 2010. Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles, in: Proc. SemEval@ACL Conf., pp. 21–26. doi:10.1007/s10579-012-9210-3.
- Kontoulis, C.G., Papagiannopoulou, E., Tsoumakas, G., 2021. Keyphrase extraction from scientific articles via extractive summarization, in: Proc. SDP@NAACL Conf., pp. 49–55. doi:10.18653/v1/2021.sdp-1.6.

Krapivin, M., Autaeu, A., Marchese, M., 2009. Large dataset for keyphrases extraction. University of Trento .

- Kuhn, H.W., 1955. The hungarian method for the assignment problem. Nav. Res. Logist. Q., 83–97doi:10.1007/978-3-540-68279-0_2.
- Kulkarni, M., Mahata, D., Arora, R., Bhowmik, R., 2022. Learning rich representation of keyphrases from text, in: Findings of the NAACL, pp. 891–906. doi:10.18653/v1/2022.findings-naacl.67.
- Lai, T.M., Bui, T., Kim, D.S., Tran, Q.H., 2020. A joint learning approach based on self-distillation for keyphrase extraction from scientific documents, in: Proc. COLING Conf., pp. 649–656. doi:10.48550/arXiv.2010.11980.
- Lancioni, G., S.Mohamed, S., Portelli, B., Serra, G., Tasso, C., 2020. Keyphrase generation with GANs in low-resources scenarios, in: Proc. SustaiNLP@EMNLP Conf., pp. 89–96. doi:10.18653/v1/2020.sustainlp-1.12.
- Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents, in: Proc. ICML Conf., pp. 1188–1196.
- Lei, Y., Hu, C., Ma, G., Zhang, R., 2021. Keyphrase extraction with incomplete annotated training data, in: Proc. W-NUT Conf., pp. 26–34. doi:10.18653/v1/2021.wnut-1.4.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L., 2020. BART: Denoising sequence-tosequence pre-training for natural language generation, translation, and comprehension, in: Proc. ACL Conf., pp. 7871–7880. doi:10.18653/ v1/2020.acl-main.703.
- Li, X., Daoutis, M., 2021. Unsupervised key-phrase extraction and clustering for classification scheme in scientific publications, in: Proc. SDU@AAAI Conf., p. 0. doi:10.48550/ARXIV.2101.09990.
- Liang, X., Wu, S., Li, M., Li, Z., 2021. Unsupervised keyphrase extraction by jointly modeling local and global context, in: Proc. EMNLP Conf., pp. 155–164. doi:10.18653/v1/2021.emnlp-main.14.
- Liang, Y., Zaki, M.J., 2021. Keyphrase extraction using neighborhood knowledge based on word embeddings. CoRR doi:10.48550/arXiv. 2111.07198.
- Liu, R., Lin, Z., Wang, W., 2021. Addressing extraction and generation separately: Keyphrase prediction with pre-trained language models. IEEE ACM Trans. Audio Speech Lang. Process., 3180–3191doi:10.1109/TASLP.2021.3120587.
- Liu, Z., Huang, W., Zheng, Y., Sun, M., 2010. Automatic keyphrase extraction via topic decomposition, in: Proc. EMNLP Conf., pp. 366-376.
- Liu, Z., Li, P., Zheng, Y., Sun, M., 2009. Clustering to find exemplar terms for keyphrase extraction, in: Proc. EMNLP Conf., pp. 257–266.
- Liu, Z., Xiong, C., Sun, M., Liu, Z., 2018. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval, in: Proc. ACL Conf., pp. 2395–2405. doi:10.18653/v1/P18-1223.
- Lopez, P., Romary, L., 2010. HUMB: Automatic key term extraction from scientific articles in GROBID, in: Proc. SemEval@ACL Conf., pp. 248–251.
- Lu, X., Chow, T.W.S., 2021. Duration modeling with semi-markov conditional random fields for keyphrase extraction. IEEE Trans. Knowl. Data Eng., 1453–1466doi:10.1109/TKDE.2019.2942295.
- Luan, Y., Ostendorf, M., Hajishirzi, H., 2017. Scientific information extraction with semi-supervised neural tagging, in: Proc. EMNLP Conf., pp. 2641–2651. doi:10.48550/arXiv.1708.06075.
- Luo, Y., Li, Z., Wang, B., Xing, X., Zhang, Q., Huang, X., 2020. Sensenet: Neural keyphrase generation with document structure. CoRR doi:10.48550/arXiv.2012.06754.
- Luo, Y., Xu, Y., Ye, J., Qiu, X., Zhang, Q., 2021. Keyphrase generation with fine-grained evaluation-guided reinforcement learning, in: Proc. Findings of EMNLP Conf., pp. 497–507. doi:10.18653/v1/2021.findings-emnlp.45.
- M, H.K., N, M.D., S, K.M., 2005. Corephrase: Keyphrase extraction for document clustering, in: Proc. MLDM Conf., pp. 265–274. doi:10.1007/ 11510888_26.
- Mahata, D., Agarwal, N., Gautam, D., Kumar, A., Parekh, S., Singla, Y.K., Acharya, A., Shah, R.R., 2022. LDKP: A dataset for identifying keyphrases from long scientific documents. CoRR doi:10.48550/arXiv.2203.15349.
- Mahata, D., Kuriakose, J., Shah, R.R., Zimmermann, R., 2018. Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings, in: Proc. NAACL, Short Papers Conf., pp. 634–639. doi:10.18653/v1/n18–2100.
- Mahfuzh, M., Soleman, S., Purwarianti, A., 2020. Improving joint layer RNN based keyphrase extraction by using syntactical features. CoRR doi:10.1109/ICAICTA.2019.8904194, arXiv:2009.07119.
- Marujo, L., Gershman, A., Carbonell, J., Frederking, R., Neto, J.P., 2012. Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization, in: Proc. LREC Conf., pp. 399–403.
- Marujo, L., Viveiros, M., Neto, J.P., 2011. Keyphrase cloud generation of broadcast news, in: Proc. INTERSPEECH Conf., pp. 2393-2396. arXiv:1306.4606.
- Medelyan, O., Frank, E., Witten, I.H., 2009. Human-competitive tagging using automatic keyphrase extraction, in: Proc. EMNLP Conf., pp. 1318–1327.
- Medelyan, O., Witten, I.H., 2006. Thesaurus based automatic keyphrase indexing, in: Proc. JCDL Conf., pp. 296–297. doi:10.1145/1141753. 1141819.
- Meng, R., Yuan, X., Wang, T., Brusilovsky, P., Trischler, A., He, D., 2019. Does order matter? an empirical study on generating multiple keyphrases as a sequence. CoRR doi:10.48550/arXiv.1909.03590.
- Meng, R., Yuan, X., Wang, T., Zhao, S., Trischler, A., He, D., 2021. An empirical study on neural keyphrase generation, in: Proc. NAACL Conf., pp. 4985–5007. doi:10.18653/v1/2021.naacl-main.396.
- Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., Chi, Y., 2017. Deep keyphrase generation, in: Proc. ACL Conf., pp. 582–592. doi:10.18653/ v1/P17-1054.
- Mihalcea, R., Tarau, P., 2004. Textrank: Bringing order into text, in: Proc. EMNLP Conf., pp. 404-411.

- Mu, F., Yu, Z., Wang, L., Wang, Y., Yin, Q., Sun, Y., Liu, L., Ma, T., Tang, J., Zhou, X., 2020. Keyphrase extraction with span-based feature representations. CoRR doi:10.48550/arXiv.2002.05407, arXiv:2002.05407.
- Nasar, Z., Jaffry, S.W., Malik, M.K., 2019. Textual keyword extraction and summarization: State-of-the-art. Inf. Process. Manag., 102088doi:10. 1016/j.ipm.2019.102088.
- Newman, M.E.J., Girvan, M., 2004. Finding and evaluating community structure in networks. Phys. Rev. E., 026113doi:10.1103/PhysRevE. 69.026113.
- Nguyen, C.Q., Phan, T.T., 2009. An ontology-based approach for key phrase extraction, in: Proc. ACL-IJCNLP, Short Papers Conf., pp. 181–184. doi:10.3115/1667583.1667639.
- Nguyen, T.D., Kan, M., 2007. Keyphrase extraction in scientific publications, in: Proc. ICADL Conf., pp. 317–326. doi:10.1007/ 978-3-540-77094-7_41.
- Nguyen, T.D., Luong, M.T., 2010. WINGNUS: Keyphrase extraction utilizing document logical structure, in: Proc. SemEval@ACL Conf., pp. 166–169.
- Nikzad-Khasmakhi, N., Feizi-Derakhshi, M., Asgari-Chenaghlu, M., Balafar, M.A., Feizi-Derakhshi, A., Rahkar-Farshi, T., Ramezani, M., Jahanbakhsh-Nagadeh, Z., Zafarani-Moattar, E., Ranjbar-Khadivi, M., 2021. Phraseformer: Multimodal key-phrase extraction using transformer and graph embedding. CoRR doi:10.48550/ARXIV.2106.04939.
- Ni'mah, I., Menkovski, V., Pechenizkiy, M., 2019. BSDAR: beam search decoding with attention reward in neural keyphrase generation. CoRR doi:10.48550/arXiv.1909.09485.
- Ohsawa, Y., Benson, N.E., Yachida, M., 1998. Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor, in: Proc. ADL Conf., pp. 12–18. doi:10.1109/ADL.1998.670375.
- Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The PageRank citation ranking: Bringing order to the web. Technical Report. Stanford InfoLab.
- Pagliardini, M., Gupta, P., Jaggi, M., 2018. Unsupervised learning of sentence embeddings using compositional n-gram features, in: Proc. ACL Conf., pp. 528–540. doi:10.18653/v1/n18-1049.
- Papagiannopoulou, E., Tsoumakas, G., 2018. Local word vectors guiding keyphrase extraction. Inf. Process. Manag., 888–902doi:10.1016/j. ipm.2018.06.004.
- Pasunuru, R., Bansal, M., 2018. Multi-reward reinforced summarization with saliency and entailment, in: Proc. ACL Conf., pp. 646–653. doi:10.18653/v1/n18-2102.
- Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation, in: Proc. EMNLP Conf., pp. 1532–1543. doi:10.3115/v1/D14-1162.
- Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations, in: Proc. NAACL Conf., pp. 2227–2237. doi:10.18653/v1/n18-1202.
- Prasad, A., Kan, M., 2019. Glocal: Incorporating global information in local convolution for keyphrase extraction, in: Proc. NAACL Conf., pp. 1837–1846. doi:10.18653/v1/n19-1182.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J., 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. , 140:1–140:67.
- Sahrawat, D., Mahata, D., Kulkarni, M., Zhang, H., Gosangi, R., Stent, A., Sharma, A., Kumar, Y., Shah, R.R., Zimmermann, R., 2019. Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings. CoRR doi:10.48550/arXiv.1910.08840, arXiv:1910.08840.
- Salton, G., Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. , 513–523doi:10.1016/0306-4573(88)90021-0.
- Santosh, T.Y.S.S., Sanyal, D.K., Bhowmick, P.K., Das, P.P., 2020a. DAKE: document-level attention for keyphrase extraction, in: Proc. ECIR Conf., pp. 392–401. doi:10.1007/978-3-030-45442-5_49.
- Santosh, T.Y.S.S., Sanyal, D.K., Bhowmick, P.K., Das, P.P., 2020b. Sasake: Syntax and semantics aware keyphrase extraction from research papers, in: Proc. COLING Conf., pp. 5372–5383. doi:10.18653/v1/2020.coling-main.469.
- Santosh, T.Y.S.S., Sanyal, D.K., Bhowmick, P.K., Das, P.P., 2021a. Gazetteer-guided keyphrase generation from research papers, in: Proc. PAKDD Conf., pp. 655–667. doi:10.1007/978-3-030-75762-5_52.
- Santosh, T.Y.S.S., Varimalla, N.R., Vallabhajosyula, A., Sanyal, D.K., Das, P.P., 2021b. Hicova: Hierarchical conditional variational autoencoder for keyphrase generation, in: Proc. CIKM Conf., pp. 3448–3452. doi:10.1145/3459637.3482119.
- Saputra, I.F., Mahendra, R., Wicaksono, A.F., 2018. Keyphrases extraction from user-generated contents in healthcare domain using long short-term memory networks, in: Proc. BioNLP Conf., pp. 28–34. doi:10.18653/v1/w18-2304.
- Sarkar, K., Nasipuri, M., Ghose, S., 2010. A new approach to keyphrase extraction using neural networks. CoRR doi:10.1109/IranianCEE. 2019.8786505.
- Schutz, A.T., et al., 2008. Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods. M. App. Sc Thesis .
- Shen, X., Wang, Y., Meng, R., Shang, J., 2022. Unsupervised deep keyphrase generation, in: Proc. AAAI Conf., pp. 11303–11311. doi:10.1609/aaai.v36i10.21381.
- Shi, T., Jiao, S., Hou, J., Li, M., 2008. Improving keyphrase extraction using wikipedia semantics, in: Proc. IITA Conf., pp. 42–46. doi:10.1109/ IITA.2008.211.
- Siddiqi, S., Sharan, A., 2015. Keyword and keyphrase extraction techniques: a literature review. International Journal of Computer Applications doi:10.5120/19161-0607.
- Song, M., Jing, L., Xiao, L., 2021. Importance estimation from multiple perspectives for keyphrase extraction, in: Proc. EMNLP Conf., pp. 2726–2736. doi:10.18653/v1/2021.emnlp-main.215.
- Sterckx, L., Demeester, T., Deleu, J., Develder, C., 2015. Topical word importance for fast keyphrase extraction, in: Proc. WWW Conf., pp. 121–122. doi:10.1145/2740908.2742730.

- Subramanian, S., Wang, T., Yuan, X., Zhang, S., Trischler, A., Bengio, Y., 2018. Neural models for key phrase extraction and question generation, in: Proc. QA@ACL Conf., pp. 78–88. doi:10.18653/v1/W18-2609.
- Sun, S., Liu, Z., Xiong, C., Liu, Z., Bao, J., 2021. Capturing global informativeness in open domain keyphrase extraction, in: Proc. NLPCC Conf., pp. 275–287. doi:10.1007/978-3-030-88483-3_21.
- Sun, Y., Qiu, H., Zheng, Y., Wang, Z., Zhang, C., 2020. Sifrank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model. IEEE Access, 10896–10906doi:10.1109/ACCESS.2020.2965087.
- Sun, Z., Tang, J., Du, P., Deng, Z., Nie, J., 2019. Divgraphpointer: A graph pointer network for extracting diverse keyphrases, in: Proc. SIGIR Conf., pp. 755–764. doi:10.1145/3331184.3331219.
- Swaminathan, A., Gupta, R.K., Zhang, H., Mahata, D., Gosangi, R., Shah, R.R., 2020a. Keyphrase generation for scientific articles using gans, in: Proc. Student Abstrct@AAAI Conf., pp. 13931–13932. doi:10.1609/aaai.v34i10.7238.
- Swaminathan, A., Zhang, H., Mahata, D., Gosangi, R., Shah, R.R., Stent, A., 2020b. A preliminary exploration of GANs for keyphrase generation, in: Proc. EMNLP Conf., pp. 8021–8030. doi:10.18653/v1/2020.emnlp-main.645.
- Teneva, N., Cheng, W., 2017. Salience rank: Efficient keyphrase extraction with topic modeling, in: Proc. ACL, Short Papers Conf., pp. 530–535. doi:10.18653/v1/P17-2084.
- Tu, Z., Lu, Z., Liu, Y., Liu, X., Li, H., 2016. Modeling coverage for neural machine translation, in: Proc. ACL Conf., pp. 76–85. doi:10.18653/ v1/p16-1008.
- Turney, P.D., 2002. Learning to extract keyphrases from text. CoRR cs.LG/0212013. doi:10.48550/ARXIV.CS/0212013.
- Vega-Oliveros, D.A., Gomes, P.S., Milios, E.E., Berton, L., 2019. A multi-centrality index for graph-based keyword extraction. Inf. Process. Manag., 102063doi:10.1016/j.ipm.2019.102063.
- Wan, X., Xiao, J., 2008. Single document keyphrase extraction using neighborhood knowledge, in: Proc. AAAI Conf., pp. 855–860. doi:10.5555/1620163.1620205.
- Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., Guo, M., 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems, in: Proc. CIKM Conf., pp. 417–426. doi:10.1145/3269206.3271739.
- Wang, J., Peng, H., Hu, J., 2005. Automatic keyphrases extraction from document using neural network, in: Proc. ICMLC Conf., pp. 633–641. doi:10.1007/11739685_66.
- Wang, L., Cardie, C., 2013. Domain-independent abstract generation for focused meeting summarization, in: Proc. ACL Conf., pp. 1395–1405.
- Wang, L., Li, S., 2017. PKU_ICL at SemEval-2017 task 10: Keyphrase extraction with model ensemble and external knowledge, in: Proc. SemEval@ACL Conf., pp. 934–937. doi:10.18653/v1/S17-2161.
- Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M., 2020a. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, in: Proc. NeurIPS Cof., pp. 5776–5788. doi:https://doi.org/10.48550/arXiv.2002.10957.
- Wang, Y., Fan, Z., Rosé, C.P., 2020b. Incorporating multimodal information in open-domain web keyphrase extraction, in: Proc. EMNLP Conf., pp. 1790–1800. doi:10.18653/v1/2020.emnlp-main.140.
- Wang, Y., Li, J., Chan, H.P., King, I., Lyu, M.R., Shi, S., 2019. Topic-aware neural keyphrase generation for social media language, in: Proc. ACL Conf., pp. 2516–2526. doi:10.18653/v1/p19-1240.
- Wang, Y., Li, J., Lyu, M., King, I., 2020c. Cross-media keyphrase prediction: A unified framework with multi-modality multi-head attention and image wordings, in: Proc. EMNLP Conf., pp. 3311–3324. doi:10.18653/v1/2020.emnlp-main.268.
- Wilson, T., Wiebe, J., Hoffmann, P., 2005. Recognizing contextual polarity in phrase-level sentiment analysis, in: Proc. HLT/EMNLP Conf., pp. 347–354.
- Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G., 1999. KEA: practical automatic keyphrase extraction, in: Proc. ACM DL Conf., pp. 254–255. doi:10.1145/313238.313437.
- Won, M., Martins, B., Raimundo, F., 2019. Automatic extraction of relevant keyphrases for the study of issue competition, in: Proc. COLING Conf., pp. 7–13. doi:10.29007/mmk4.
- Wu, D., Ahmad, W.U., Dev, S., Chang, K., 2022a. Representation learning for resource-constrained keyphrase generation. CoRR doi:10.48550/ arXiv.2203.08118.
- Wu, H., Liu, W., Li, L., Nie, D., Chen, T., Zhang, F., Wang, D., 2021. Unikeyphrase: A unified extraction and generation framework for keyphrase prediction, in: Proc. Findings of ACL-IJCNLP Conf., pp. 825–835. doi:10.18653/v1/2021.findings-acl.73.
- Wu, H., Ma, B., Liu, W., Chen, T., Nie, D., 2022b. Fast and constrained absent keyphrase generation by prompt-based learning, in: Proc. AAAI Conf., pp. 11495–11503. doi:10.1609/aaai.v36i10.21402.
- Xie, B., Wei, X., Yang, B., Lin, H., Xie, J., Wang, X., Zhang, M., Su, J., 2022. Wr-one2set: Towards well-calibrated keyphrase generation, in: Proc. EMNLP Conf., pp. 7283–7293.
- Xie, F., Wu, X., Zhu, X., 2017. Efficient sequential pattern mining with wildcards for keyphrase extraction. Knowl. Based Syst., 27–39doi:10.1016/j.knosys.2016.10.011.
- Xiong, L., Hu, C., Xiong, C., Campos, D., Overwijk, A., 2019. Open domain web keyphrase extraction beyond language modeling, in: Proc. EMNLP Conf., pp. 5174–5183. doi:10.48550/arXiv.1911.02671.
- Ye, H., Wang, L., 2018. Semi-supervised learning for neural keyphrase generation, in: Proc. EMNLP Conf., pp. 4142–4153. doi:10.18653/v1/p19-1515.
- Ye, J., Cai, R., Gui, T., Zhang, Q., 2021a. Heterogeneous graph neural networks for keyphrase generation, in: Proc. EMNLP Conf., pp. 2705–2715. doi:10.18653/v1/2021.emnlp-main.213.
- Ye, J., Gui, T., Luo, Y., Xu, Y., Zhang, Q., 2021b. One2set: Generating diverse keyphrases as a set, in: Proc. ACL Conf., pp. 4598–4608. doi:10.18653/v1/2021.acl-long.354.
- Yih, W., Goodman, J., Carvalho, V.R., 2006. Finding advertising keywords on web pages, in: Proc. WWW Conf., pp. 213–222. doi:10.1145/ 1135777.1135813.
- Yuan, X., Wang, T., Meng, R., Thaker, K., Brusilovsky, P., He, D., Trischler, A., 2020. One size does not fit all: Generating and evaluating variable

number of keyphrases, in: Proc. ACL Conf., pp. 7961–7975. doi:10.18653/v1/2020.acl-main.710.

- Zhang, C., 2008. Automatic keyword extraction from documents using conditional random fields. J. Comput. Inf. Syst., 1169–1180.
- Zhang, K., Xu, H., Tang, J., Li, J., 2006. Keyword extraction using support vector machine, in: Proc. WAIM Conf., pp. 85–96. doi:10.1007/11775300\ 8.
- Zhang, L., Chen, Q., Wang, W., Deng, C., Zhang, S., Li, B., Wang, W., Cao, X., 2021. Mderank: A masked document embedding rank approach for unsupervised keyphrase extraction. CoRR doi:10.48550/ARXIV.2110.06651.
- Zhang, Q., Wang, Y., Gong, Y., Huang, X., 2016. Keyphrase extraction using deep recurrent neural networks on twitter, in: Proc. EMNLP Conf., pp. 836–845. doi:10.18653/v1/d16-1080.
- Zhang, Y., Chang, Y., Liu, X., Gollapalli, S.D., Li, X., Xiao, C., 2017a. MIKE: keyphrase extraction by integrating multidimensional information, in: Proc. CIKM Conf., pp. 1349–1358. doi:10.1145/3132847.3132956.
- Zhang, Y., Fang, Y., Xiao, W., 2017b. Deep keyphrase generation with a convolutional sequence to sequence model, in: Proc. ICSAI Conf., pp. 1477–1485. doi:10.1109/ICSAI.2017.8248519.
- Zhang, Y., Jiang, T., Yang, T., Li, X., Wang, S., 2022. HTKG: deep keyphrase generation with neural hierarchical topic guidance, in: Proc. SIGIR Conf., pp. 1044–1054. doi:10.1145/3477495.3531990.
- Zhang, Y., Li, J., Song, Y., Zhang, C., 2018. Encoding conversation context for neural keyphrase extraction from microblog posts, in: Proc. NAACL Conf., pp. 1676–1686. doi:10.18653/v1/n18-1151.
- Zhang, Y., Zhang, C., 2019. Using human attention to extract keyphrase from microblog post, in: Proc. ACL Conf., pp. 5867–5872. doi:10.18653/ v1/p19-1588.
- Zhang, Y., Zincir-Heywood, A.N., Milios, E.E., 2004. World wide web site summarization. Web Intell. Agent Syst., 39–53doi:10.5555/1039791.1039794.
- Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q., 2019. ERNIE: enhanced language representation with informative entities, in: Proc. ACL Conf., pp. 1441–1451. doi:10.18653/v1/p19–1139.
- Zhao, J., Bao, J., Wang, Y., Wu, Y., He, X., Zhou, B., 2021. SGG: learning to select, guide, and generate for keyphrase generation, in: Proc. NAACL Conf., pp. 5717–5726. doi:10.18653/v1/2021.naacl-main.455.
- Zhao, J., Zhang, Y., 2019. Incorporating linguistic constraints into keyphrase generation, in: Proc. ACL Conf., pp. 5224–5233. doi:10.18653/v1/ P19-1515.
- Zhuang, L., Wayne, L., Ya, S., Jun, Z., 2021. A robustly optimized BERT pre-training approach with post-training, in: Proc. CCL Conf., pp. 471–484. doi:10.1007/978-3-030-84186-7_31.



Binbin Xie received the Bachelor degree in the school of informatics, Xiamen University, in 2021. And she is studying for a master's degree under the supervision of Prof. Jinsong Su now. Her research interests include code generation, keyphrase generation and machine translation.



Jia Song was born in 2000. She received her Bachelor degree in Economic Information Engineering School from Southwest University of Finance and Economics, and is a graduate student under the supervision of Prof. Jinsong Su now. Her major research interests are natural language processing and keyphrase generation.



Liangying Shao was born in 2000. She received her Bachelor degree in the school of informatics, Xiamen University, and is a graduate student under the supervision of Prof. Jinsong Su now. Her major research interests are natural language processing and keyphrase generation.



Suhang Wu was born in 2000. He is a undergraduate student at the College of Computer Science and Electronic Engineering of Hunan University now. He will become a graduate student at Xiamen University under the supervision of Prof. Jinsong Su.



Xiangpeng Wei received the Ph.D. degree from the University of Chinese Academy of Sciences (UCAS) in 2021, supervised by Prof. Yue Hu. He is now a senior algorithm engineer in the Language Technology Lab at Alibaba DAMO Academy. His research interests include natural language processing and neural machine translation.



Baosong Yang received the Ph.D at NLP2CT Lab of University of Macau, advised by Prof. Derek F. Wong, and is currently an algorithm expert in the Language Technology Lab at Alibaba DAMO Academy. His research interests include natural language processing and machine translation.



Huan Lin received a master's degree in Xiamen University supervised by Prof. Jinsong Su, and is now an algorithm engineer in the Language Technology Lab at Alibaba DAMO Academy. Her research interests include natural language processing and machine translation.



Jun Xie received the Ph.D. degree in computer science from the Chinese Academy of Sciences, Beijing, China. He is currently a senior staff algorithm engineer in the Language Technology Lab at Alibaba DAMO Academy. His research interests include natural language processing and machine translation.



Jinsong Su was born in 1982. He received the Ph.D. degree in Chinese Academy of Sciences, and is now a professor in Xiamen University. His research interests include natural language processing and machine translation.