A Survey on Recent Advances in Keyphrase Extraction from Pre-trained Language Models

Mingyang Song, Yi Feng and Liping Jing*
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University
Beijing, China
{mingyang.song,21112027,lpjing}@bjtu.edu.cn

Abstract

Keyphrase Extraction (KE) is a critical component in Natural Language Processing (NLP) systems for selecting a set of phrases from the document that could summarize the important information discussed in the document. Typically, a keyphrase extraction system can significantly accelerate the speed of information retrieval and help people get first-hand information from a long document quickly and accurately. Specifically, keyphrases are capable of providing semantic metadata characterizing documents and producing an overview of the content of a document. In this paper, we introduce keyphrase extraction, present a review of the recent studies based on pre-trained language models, offer interesting insights on the different approaches, highlight open issues, and give a comparative experimental study of popular supervised as well as unsupervised techniques on several datasets. To encourage more instantiations, we release the related files mentioned in this paper[1].

1 Introduction

Keyphrase extraction is a fundamental task in NLP for identifying and extracting a set of keyphrases from the document that could summarize the important information discussed in the source document (Hasan and Ng, 2014; Papagiannopoulou and Tsoumakas, 2019). Keyphrases have enabled accurate and fast searching for the document from a large text corpus and have exhibited their potential in improving many NLP tasks, such as text summarization (Zhang et al., 2004). Various information filtering and extracting techniques are becoming critical with the ever-increasing amount of text data. Owing to its potential importance, keyphrase extraction has received more and more attention from NLP researchers. However, the keyphrase extraction task is far from being solved: state-of-the-art performance on keyphrase extraction is still lower than other core NLP tasks. Our goal in this paper is to investigate the state-of-the-art models in keyphrase extraction, examine the primary sources of errors made by existing systems, and discuss the challenges ahead.

The first keyphrase extraction task was organized by Turney (1999), which defines the keyphrase extraction task as "the automatic selection of important and topical phrases from the body of a document". Since then, there have been numerous keyphrase extraction models (Witten et al., 1999; Turney, 2000; Tomokiyo and Hurst, 2003; Hulth, 2004; Wan and Xiao, 2008a; Jiang et al., 2009; Liu et al., 2009; Grineva et al., 2009; Nguyen and Phan, 2009; Bougouin et al., 2013; Caragea et al., 2014; Danesh et al., 2015; Bougouin et al., 2016; Florescu and Caragea, 2017a; Campos et al., 2018a; Alzaidy et al., 2019). In the past two decades, keyphrase extraction methods have experienced the development from traditional approaches to deep learning methods (Hasan and Ng, 2014; Papagiannopoulou and Tsoumakas, 2019). With the recent development of Pre-trained Language Models (PLMs) (Devlin et al., 2019; Liu et al., 2019), many NLP tasks have significantly changed, that is, how to adopt and leverage pre-trained language models in the specific task. Therefore, many keyphrase extraction models (Sun et al., 2020a; Song et al., 2021) adopt PLMs as the embedding layer.

We present a comprehensive survey of recent advances in neural keyphrase extraction. We describe the neural keyphrase extraction systems based on pre-trained language models, which depend on different paradigms (e.g., one-stage (Wang et al., 2020) and two-stage (Sun et al., 2020a)), various tasks (e.g., classification and ranking (Mu et al., 2020; Sun et al., 2020a)), different learning strategies (e.g., supervised (Song et al., 2021) and un-

*Corresponding author.
[1] https://github.com/MySong7NLPer/KeyphraseExtractionSurvey

Findings of the Association for Computational Linguistics: EACL 2023, pages 2108–2119
May 2-6, 2023

supervised (Ding and Luo, 2021)), and variants of pre-trained language models (e.g., BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)).

Furthermore, we re-implement and collect the results of the mentioned models on several benchmark keyphrase extraction datasets. We illustrate the results in Table 3 and Table 2 and discuss in Section 6 how neural keyphrase extraction systems have improved performance over past works, including supervised and unsupervised models. Furthermore, we provide resources, including links to share the current neural keyphrase extraction systems and links to share the code for each category of the neural keyphrase extraction approaches. To the best of our knowledge, this is the first survey focusing on the keyphrase extraction task based on recent pre-trained language models.

Overall, this paper first discusses previous surveys on keyphrase extraction in Section 2.1 and give a briefly introduction about pre-trained language models in Section 2.2. Then we highlight standard, past, and recent benchmark keyphrase extraction datasets (from shared tasks and other research) in Section 3 and evaluation metrics in Section 4. We then describe neural keyphrase extraction systems in Section 5. Next, we give the analysis and discussion in Section 6. Finally, we summarize the conclusions and future directions of neural keyphrase extraction in Section 7. The limitations of our work is presented in Section 8.

2 Preliminary

In this section, we claim the differences between the current survey and the existing surveys. Next, we present the background of pre-trained language models and their importance in NLP.

2.1 Previous Surveys

The first comprehensive keyphrase extraction survey was Hasan and Ng (2014), which covered a variety of unsupervised and supervised keyphrase extraction models, highlighted common features used by existing models during that time, and explained evaluation metrics that are still in use today. Papagiannopoulou and Tsoumakas (2019) present a more recent keyphrase extraction survey that mainly included many unsupervised and supervised models based on deep learning. Furthermore, Papagiannopoulou and Tsoumakas (2019) also provides a list of popular keyphrase extraction datasets and a thorough empirical study.

The existing keyphrase extraction surveys primarily cover early feature-engineered and neural-based keyphrase extraction models (Hasan and Ng, 2014; Papagiannopoulou and Tsoumakas, 2019). There is not yet, to our knowledge, a comprehensive survey of keyphrase extraction based on pre-trained language models.

2.2 Pre-trained Language Models

Recently, pre-trained language models have advanced the state-of-the-art in many NLP tasks ranging from textual similarity to text summarization (Zhang et al., 2019; Liu and Lapata, 2019; Zhong et al., 2020) and named entity recognition (Zhou et al., 2021). State-of-the-art pre-trained models include LSTM-based language models (e.g., ELMo (Peters et al., 2018)) and Transformer-based language models (e.g., BERT[2] (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)). Specifically, the transformer-based models learn bidirectional representations for words based on a masked language model and sentence adjacency training objective (Devlin et al., 2019). Simply using contextualized embeddings obtained from the transformer-based pre-trained language models in place of traditional embeddings has resulted in state-of-the-art performance on a range of NLP tasks. Therefore, pre-trained language models have been employed as encoders for obtaining word-, sentence-, and document-level representations to assist the downstream tasks.

3 Keyphrase Extraction Dataset

Since the first shared task on KE (Turney, 1999), many shared tasks and benchmark datasets for KE have been created. Specifically, OpenKP (Xiong et al., 2019), Inspec (Hulth, 2003), NUS (Nguyen and Kan, 2007), Krapivin (Krapivin and Marchese, 2009), SemEval2010 (Kim et al., 2010), SemEval2017 (Augenstein et al., 2017), and KP20k (Meng et al., 2017) were created from scientific articles in English.

Compared with other datasets, KP20k contains a large amount of annotation data, so it is often used as the dataset to train the neural-based KE models recently. Meanwhile, in recent papers (Sun et al., 2020a; Song et al., 2021), Inspec (Hulth, 2003), NUS (Nguyen and Kan, 2007), Krapivin (Krapivin and Marchese, 2009), SemEval2010 (Kim et al., 2010), and SemEval2017 (Augenstein et al., 2017)

[2] https://huggingface.co/bert-base-cased

Dataset	Type	Long	# Doc.	Avg. # Words	Present KPs (%)
KP20k (Meng et al., 2017)	Scientific Paper Abstract	✗	568.00k	188.47	57.40
Inspec (Hulth, 2003)	Scientific Paper Abstract	✗	2.00k	130.57	55.69
SemEval2017 (Augenstein et al., 2017)	Scientific Paper Abstract	✗	0.50k	176.13	42.01
NUS (Nguyen and Kan, 2007)	Full Scientific Paper	✓	0.21k	7644.43	67.75
Krapivin (Krapivin and Marchese, 2009)	Full Scientific Paper	✓	2.30k	8420.76	44.74
SemEval2010 (Kim et al., 2010)	Full Scientific Paper	✓	0.24k	7434.52	88.70
DUC2001 (Wan and Xiao, 2008b)	News Document	✗	0.31k	724.63	97.82
OpenKP (Xiong et al., 2019)	Open Domain Web Content	✗	147.20k	900.40	100.00

Table 1: This table shows the statistics of different recent popular datasets. **Long** indicates whether the dataset belongs to a long document. **# Doc.** is the number of documents in the dataset. **Avg. # Words** is the average number of words for documents in the indicated dataset. **Present KPs (%)** indicates the percentage of keyphrases, which are presented in the documents.

datasets are often used as the zero-shot test sets to verify the robustness of the KE models trained by the KP20k dataset. Furthermore, KE tasks have also been organized on newswire articles in English, e.g., DUC2001 (Wan and Xiao, 2008b). Table 1 summarizes the statistics of several commonly used benchmark datasets.

4 Keyphrase Extraction Evaluation

This section describes evaluation metrics for measuring recent state-of-the-art keyphrase extraction baselines on commonly-used datasets. Designing a suitable evaluation metric for the keyphrase extraction task is by no means an easy study (Hasan and Ng, 2014). To score the output of a keyphrase extraction model, the traditional approach, which is also adopted by the SemEval-2010 (Kim et al., 2010) shared task on keyphrase extraction, is (1) to create a mapping between the keyphrases in the ground-truth keyphrases and those in the model output adopting exact and partial matching (Papagiannopoulou and Tsoumakas, 2019), and then (2) score the output using evaluation metrics such as precision (P), recall (R), and F1-score (F1).

As mentioned earlier, such evaluation usually operates based on exact matches between the predicted and ground-truth keyphrases. However, such a strategy cannot account for partial matches or semantic similarity. For example, if the prediction is "keyphrase extraction model" and the ground truth is "keyphrase extraction system", despite both semantic similarity and partial matching, the score will be 0. These minor deviations are ubiquitous in keyphrase extraction, yet they are harshly penalized by the "exact match" evaluation metrics.

5 Neural Keyphrase Extraction Models with Pre-trained Language Models

There are two popular pipelines in the keyphrase extraction task, including one-stage and two-stage frameworks, as illustrated in Figure 1. The former mainly refers to using the task reformulation to address the keyphrase extraction task, which often treats the keyphrase extraction task as a sequence labeling task. The latter represents a more general framework, which usually operates in two procedures: (1) extracting a set of words/phrases that serve as candidate phrases using some heuristics and (2) determining which candidate phrases are keyphrases using supervised or unsupervised methods (Hasan and Ng, 2014; Papagiannopoulou and Tsoumakas, 2019).

Typically, supervised methods perform better on specific domain tasks. However, this kind of method takes a lot of labor to annotate the corpus, and the model after training may overfit and not work well on other KE datasets. On the contrary, unsupervised methods do not need to annotate the corpus and usually have better data generalization in different domains. Still, the performance is often insufficient due to the lack of annotated data. Overall, we defined the above two procedures as the candidate keyphrase extraction and keyphrase importance estimation. In this paper, we distinguish the existing methods into three categories depending on the recent state-of-the-art baselines (with

Figure 1: The overall architecture of the two-stage supervised and unsupervised keyphrase extraction framework.

pre-trained language models as the backbone), including two-stage unsupervised, two-stage supervised, and one-stage supervised models.

5.1 Two-Stage Unsupervised Keyphrase Extraction Models

As noted before, unsupervised keyphrase extraction systems generally extract a set of phrases from the source document as candidates by using heuristic rules. These rules are designed to avoid spurious instances and keep the number of candidates to a minimum (Hasan and Ng, 2014). The main steps of the commonly used candidate keyphrases extraction methods for the recent unsupervised keyphrase extraction models are as follows, (1) tokenizing the document and tagging the document with part-of-speech (POS) tags via the StanfordCoreNLP Tools[3]; (2) extracting candidate phrases based on part-of-speech tags by the regular expression via the python package NLTK[4]. Furthermore, different pruning heuristics have been designed for pruning candidates that are unlikely to be keyphrases to obtain a better candidate set (Huang et al., 2006; Kumar and Srinathan, 2008; El-Beltagy and Rafea, 2009; Newman et al., 2012; You et al., 2009). After obtaining candidates, keyphrases are determined by estimating the importance of each candidate through various strategies. Here, to facilitate the introduction, we divide the methods of importance estimation into two categories, namely, traditional methods and embedding-based methods.

Traditional unsupervised keyphrase extraction systems can be mainly divided into statistics-based (Jones, 2004; Campos et al., 2018b), topic-based (Liu et al., 2009; Jardine and Teufel, 2014), and graph-based (Mihalcea and Tarau, 2004; Wan and Xiao, 2008b; Bougouin et al., 2013; Florescu and Caragea, 2017b) methods. Generally, these models primarily use different features of documents (e.g., word frequency, position, linguistic properties, topic, length, the relationship between words, external knowledge-based information, etc.) to estimate the importance of each candidate phrase and discriminate whether a candidate phrase is a keyphrase (Hasan and Ng, 2014; Papagiannopoulou and Tsoumakas, 2019).

However, these traditional unsupervised models estimate the importance scores of candidate phrases based on the surface-level features, ignoring the high-level features (e.g., syntactic and semantic information) of natural languages, which leads to extract wrong keyphrases. Therefore, recent studies focus on embedding-based models (Wang et al., 2015; Mahata et al., 2018a; Papagiannopoulou and Tsoumakas, 2018; Sahrawat et al., 2020; Kulkarni et al., 2022; Song et al., 2022b), which leverage pre-trained embeddings (containing high-level features) to obtain phrase and document embeddings and calculate the importance scores of candidate phrases for extracting keyphrases. Wang et al. (2015) is the first work to explore utilizing word embedding and frequency to generate weighted edges between words, then using the weighted PageRank algorithm to compute and rank candidate scores. Key2vec (Mahata et al., 2018a) proposes an effective way of processing text documents for training multi-word phrase embeddings that are used for topic representations of scientific articles and ranking of keyphrases extracted from them using the topic-weighted PageRank algorithm. Mahata et al. (2018b) uses a combination of theme-weighted personalized PageRank algorithm and neural phrase embeddings for extracting and ranking keyphrases. EmbedRank (Bennani-Smires et al., 2018) ranks candidate phrases by measuring the semantic similarity between each candidate phrase and document embeddings.

With the development of pre-trained language

[3] https://stanfordnlp.github.io/CoreNLP
[4] https://github.com/nltk

models (e.g., ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), and RoBERTa (Liu et al., 2019)), SIFRank[5] (Sun et al., 2020b) improves candidate phrase and document embeddings from EmbedRank with the pre-trained language model ELMo (Peters et al., 2018) and achieves better performance. JointGL[6] (Liang et al., 2021) integrates boundary-aware phrase centrality (the semantic similarities are calculated between all candidate phrases for identifying which candidate is better) and phrase-document relevance (the semantic similarities are calculated between candidate phrases and their corresponding document) from both local and global views, then used both jointly to determine the importance of each candidate. AttentionRank[7] (Ding and Luo, 2021) adopts a pre-trained language model to calculate the self-attention of a candidate within the context of a sentence, and the cross-attention between a candidate and sentences within the source document to evaluate the local and global importance of each candidate. MDERank[8] (Zhang et al., 2021) proposes to rank candidates using the similarity between the BERT embeddings of the source document and the masked document. Totally, these models achieve state-of-the-art performance in the unsupervised keyphrase extraction task, benefiting from the development of representation learning.

5.2 Two-Stage Supervised Keyphrase Extraction Models

Different from two-stage unsupervised approaches, supervised approaches generally combine candidate keyphrase extraction and keyphrase importance estimation via *an end-to-end learning framework*, guide the whole model to rank and extract keyphrases through annotated data and optimize the two stages simultaneously. Therefore, to obtain sufficient candidates, the recent supervised models (Xiong et al., 2019; Sun et al., 2020a; Song et al., 2021, 2022a) directly extract n-grams from the document as candidates. Then propose, various approaches to estimate the importance scores of candidates. To estimate the importance of candidate phrases, similar to unsupervised models, supervised models (Xiong et al., 2019; Sun et al., 2020a; Song et al., 2021) also obtain phrase and document representations by adopting pre-trained language models as the backbone, including ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), etc.

Firstly, BLING-KPE (Xiong et al., 2019) formulates keyphrase extraction as an n-gram level keyphrase chunking task to determine whether a candidate is a keyphrase, which incorporates pre-trained embeddings (i.e., ELMo (Peters et al., 2018)) into a convolutional transformer network to model n-gram representations. BLING-KPE achieves significant improvement over previous models. To leverage external knowledge to assist keyphrase extraction, SMART-KPE[9] (Wang et al., 2020) also shows that incorporating multimodal information in web pages, such as font, size, and DOM features, can bring further improvement for open-domain web keyphrase extraction. Later, Ainslie et al. (2020) replaces the full self-attention of Transformers with local-global attention, which significantly boosts the keyphrase extraction performance for long documents. SKE-BASE-RANK (Mu et al., 2020) proposes a span-based keyphrase extraction model to model the relationships between candidates and the document in context.

JointKPE[10] (Sun et al., 2020a) proposes an open-domain keyphrase extraction approach built on pre-trained language models (Devlin et al., 2019; Liu et al., 2019), which can capture both local phraseness and global informativeness when extracting keyphrases. JointKPE learns to rank keyphrases by estimating their informativeness in the whole document and is jointly trained on the keyphrase chunking task to guarantee the phraseness of keyphrase candidates. KIEMP[11] (Song et al., 2021) proposes estimating the importance score of each candidate from multiple perspectives and introducing a matching module to match the high-level concept between the document and candidates to enhance the relevance of extracted keyphrases. To extract more relevant keyphrases, HyperMatch[12] (Song et al., 2022a) proposes a new matching framework and explores keyphrase extraction in the hyperbolic space. Concretely, HyperMatch first maps phrase and document representations into the same hyperbolic space and explicitly models the relevance between candidate phrases and the document as the phrase-document relevance via the Poincaré distance to extract keyphrases.

[5] https://github.com/sunyilgdx/SIFRank
[6] https://github.com/xnliang98/uke_ccrank
[7] https://github.com/hd10-iupui/AttentionRank
[8] https://github.com/linhanz/mderank
[9] https://github.com/victorywys/SMART-KPE
[10] https://github.com/thunlp/BERT-KPE
[11] https://github.com/MySong7NLPer/KIEMP
[12] https://github.com/MySong7NLPer/HyperMatch

Model	DUC2001			Inspec			SemEval2010			SemEval2017		
	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15
Traditional Two-Stage Models												
TF-IDF (Jones, 2004)	9.21	10.63	11.06	11.28	13.88	13.83	2.81	3.48	3.91	12.70	16.26	16.73
YAKE (Campos et al., 2018b)	12.27	14.37	14.76	18.08	19.62	20.11	11.76	14.4	15.19	11.84	18.14	20.55
TextRank (Mihalcea and Tarau, 2004)	11.80	18.28	20.22	27.04	25.08	36.65	3.80	5.38	7.65	16.43	25.83	30.50
SingleRank (Wan and Xiao, 2008b)	20.43	25.59	25.70	27.79	34.46	36.05	5.90	9.02	10.58	18.23	27.73	31.73
TopicRank (Bougouin et al., 2013)	21.56	23.12	20.87	25.38	28.46	29.49	12.12	12.90	13.54	17.10	22.62	24.87
PositionRank (Florescu and Caragea, 2017b)	23.35	28.57	28.60	28.12	32.87	33.32	9.84	13.34	14.33	18.23	26.30	30.55
Two-Stage Embedding-based Unsupervised Keyphrase Extraction Models with Static Embeddings												
EmbedRankd2v (Bennani-Smires et al., 2018)	24.02	28.12	28.82	31.51	37.94	37.96	3.02	5.08	7.23	20.21	29.59	33.94
KeyGames (Saxena et al., 2020)	24.42	28.28	29.77	32.12	**40.48**	40.94	11.93	14.35	14.62	-	-	-
Two-Stage Embedding-based Unsupervised Keyphrase Extraction Models with PLMs												
SIFRank (Sun et al., 2020b)	24.27	27.43	27.86	29.11	38.80	39.59	-	-	-	22.59	32.85	38.10
JointGL (Liang et al., 2021)	28.62	**35.52**	**36.29**	**32.61**	40.17	**41.09**	13.02	**19.35**	**21.72**	-	-	-
AttentionRank (Ding and Luo, 2021)	-	-	-	24.45	32.15	34.49	11.39	15.12	16.66	**23.59**	**34.37**	**38.21**
MDERank (Zhang et al., 2021)	23.31	26.65	26.42	27.85	34.36	36.40	**13.05**	18.27	20.35	20.37	31.21	36.63

Table 2: Performance of unsupervised keyphrase extraction models on the DUC2001, Inspec, SemEval2010 and SemEval2017 test sets. F1 scores on the top 5, 10, and 15 keyphrases are reported. The best results are bolded. The results of baseline models are those presented in the original papers or better results published in other papers recently.

5.3 One-Stage Supervised Keyphrase Extraction Models

A major limitation of the above two-stage supervised approaches is classifying the labels of each candidate phrase independently while ignoring the dependencies that could potentially exist between candidates. Therefore, recent studies (Gollapalli et al., 2017; Basaldella et al., 2018; Wang et al., 2018; Alzaidy et al., 2019; Sun et al., 2019; Mu et al., 2020; Sahrawat et al., 2020) formulated keyphrase extraction as sequence labeling and showed that using linear-chain Conditional Random Fields improved the performance over baseline models for this task. Then, Mu et al. (2020) proposes SKE-BASE-CLS and -RANK, which directly extracts span-based phrase representations from all the document tokens via pre-trained language models and further learn to capture the interaction between them and their corresponding document to get better ranking results. Furthermore, this kind of model can extract overlapped keyphrases (Mu et al., 2020).

6 Discussion

In this section, we report the results of the recent unsupervised and supervised keyphrase extraction baselines, which all adopt pre-trained language models as the backbone, as shown in Table 2 and Table 3. Specifically, Table 2 presents the results of the traditional unsupervised methods and the unsupervised embedding-based keyphrase extraction baselines discussed in Section 5.1 on the DUC2001 (Wan and Xiao, 2008b), Inspec (Hulth, 2003), SemEval2010 (Kim et al., 2010), and SemEval2017 (Augenstein et al., 2017) datasets. Embedding-based two-stage models without PLMs indicate that the models do not use pre-trained language models as the backbone to obtain representations. Table 3 shows the results of all the different categories of the supervised keyphrase extraction systems discussed in Section 5.2 and Section 5.3 on the KP20k (Meng et al., 2017) and OpenKP (Xiong et al., 2019) datasets.

Our first finding from the survey is those two-stage embedding-based systems with static embeddings outperform two-stage traditional methods, despite the latter's access to different valuable features (e.g., word frequency, position, linguistic properties, topic, length, the relationship between words, external knowledge-based information, etc.). This further demonstrates the necessity of studying embedding-based methods.

Our second finding is those embedding-based systems with PLMs outperform embedding-based approaches with static embeddings in most cases.

Model	KP20k		OpenKP		
	F1@5	F1@10	F1@1	F1@3	F1@5
One-Stage Supervised Keyphrase Extraction Models					
SMART-KPE+Full (Wang et al., 2020)	-	-	**38.0**	**40.1**	**34.4**
BERT-TagKPE†	38.8	31.7	32.1	36.1	31.4
BERT-SpanKPE†	36.8	30.8	31.8	33.2	28.9
RoBERTa-TagKPE‡	39.3	32.0	36.1	38.0	33.0
RoBERTa-SpanKPE‡	37.3	30.9	34.7	36.1	31.3
Two-Stage Supervised Keyphrase Extraction Models					
BLING-KPE (Xiong et al., 2019)	-	-	26.7	29.2	20.9
SKE-BASE-CLS (Mu et al., 2020)	38.6	32.6	-	-	-
BERT-ChunkKPE†	41.2	33.7	34.0	35.6	31.1
RoBERTa-ChunkKPE†	40.8	33.7	35.5	37.3	32.4
SKE-BASE-RANK (Mu et al., 2020)	39.2	33.0	-	-	-
BERT-RankKPE†	41.3	34.0	34.2	37.4	32.5
RoBERTa-RankKPE†	41.7	34.3	36.1	39.0	33.7
HyperMatch (Song et al., 2022a)	41.6	34.3	36.4	39.4	33.8
BERT-JointKPE†	41.1	33.8	34.9	37.6	32.5
RoBERTa-JointKPE†	41.9	34.4	36.4	39.1	33.8
KIEMP (Song et al., 2021)	**42.1**	**34.5**	36.9	39.2	34.0

Table 3: Results of different categories of supervised keyphrase extraction models on two benchmark keyphrase datasets. F1 scores on the top $1, 3, 5$, and 10 keyphrases are reported. † indicates the results are reported by their corresponding paper (Sun et al., 2020a), and ‡ denotes that these results are re-evaluated by ourselves via the code which is provided by its corresponding paper (Sun et al., 2020a). The best results are highlighted in bold. The results of baseline models are those presented in the original papers or better results published in other papers recently.

However, not all embedding-based systems with PLMs are superior to embedding-based systems with static embeddings. The former generally outperforms the latter when adopting the same importance estimation strategy, but the estimation strategy can significantly affect the results of keyphrase extraction. To sum up, effectively using pre-trained embeddings to estimate the importance score of each candidate is a critical part of improving the performance of keyphrase extraction. Furthermore, there is still interesting progress to be made by leveraging a self-supervised learning strategy to optimize embedding-based systems. MDERank uses a simple yet effective contrastive learning strategy to optimize embedding-based systems, achieving better performance.

Our third finding is that the embedding-based methods have slight improvement on long document datasets (e.g., SemEval2010), and all unsupervised methods have poor effects on long document datasets. This demonstrates that keyphrase extraction from long documents is still a challenging problem.

Our final finding is that two-stage supervised keyphrase extraction methods are superior to one-stage supervised keyphrase extraction methods, as illustrated in Table 3. In addition, the two-stage method has higher scalability and adaptability than the one-stage method, such as handling long and extremely long documents.

7 Conclusion and Future Directions

We summarize the recent neural keyphrase extraction models based on pre-trained language models. Our survey of models for keyphrase extraction, covering both unsupervised and supervised models, has yielded several important insights. The analysis revealed that there are at least six major challenges ahead.

7.1 Improving the Quality of Generated Candidate Keyphrases

Many heuristic rules have proven effective with a high recall to cover most of the gold keyphrases

of source documents, which determines the upper bound of the performance of keyphrase extraction (Hasan and Ng, 2014). Intuitively, better candidate keyphrase extraction strategies are required to generate a set of candidate keyphrases with a higher recall from the source document to improve the upper-bound performance of keyphrase extraction. Recent work (Jawahar et al., 2019) demonstrates that the intermediate layers of BERT encode a rich hierarchy of linguistic information, with surface features at the bottom, syntactic features in the middle, and semantic features at the top, as mentioned in Section 2.2. They also observe that BERT mostly captures phrase-level information in the lower layers and gradually dilutes this information in higher layers. In addition, the number of candidate keyphrases will increase as the document length increases. Therefore, how constructing candidate keyphrases using the potential knowledge of pre-trained language models is a valuable research direction.

7.2 Improving Evaluation Metric

As mentioned in Section 4, the existing evaluation metrics occur when a keyphrase extraction system extracts a keyphrase from candidates that is semantically equivalent to a ground-truth keyphrase but is considered erroneous by a scoring function because it fails to recognize that the predicted keyphrase and the corresponding gold keyphrase are semantically equivalent.

In other words, an evaluation error is not made by a keyphrase extraction system, but a mistake due to an unformed scoring function (Hasan and Ng, 2014). Therefore, a more suitable evaluation metric is required to evaluate the predicted keyphrases by adopting the semantic-based matching metric instead of the exact matching evaluation metric. In the future, using pre-trained language models (e.g., BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019)) to construct a new semantic-aware evaluation metric similar to BERTScore (Zhang et al., 2020) may be an interesting and valuable research direction.

7.3 Reducing Over-Generation Error

Over-generation errors occur when a keyphrase extraction system correctly predicts a candidate as a keyphrase because it contains a word that frequently appears in the associated document but at the same time erroneously outputs other candidates as keyphrases because they have the same word in the document.

As mentioned before, for example, if the prediction is "keyphrase extraction challenge" and the ground truth is "keyphrase extraction system", despite both semantic similarity and partial matching, the score will be 0. These minor deviations are ubiquitous in keyphrase extraction, yet they are harshly penalized by the "exact match" evaluation metrics. There are often some non-keyphrases in the candidates. Half of the content of such phrases is very relevant to the core information of the document, but the other half is meaningless. These candidate keyphrases are usually hard to extract and treated as hard samples, which is one of the main reasons for reducing keyphrase extraction performance. The above issues can be solved by modifying the traditional evaluation metrics with semantic weighting.

7.4 Handling Long Document

Generally, two main challenges exist in keyphrase extraction systems equipped with pre-trained language models (e.g., BERT (Devlin et al., 2019)) as the backbone when extracting keyphrases from a long document, especially for an extremely long document.

The first challenge is that pre-trained language models can not directly model the complete context information when facing long documents due to the length limitation of pre-trained language models.

The second challenge is that as the length of the document increases, the difficulty of estimating the importance scores of candidate phrases also increases (specifically for the number of candidates), resulting in the reduction of keyphrase extraction accuracy.

7.5 Improving Domain Generalization

For news or scientific documents, the authors usually annotate a set of keyphrases for their articles (Meng et al., 2017; Augenstein et al., 2017). However, there is typically a lack of keyphrases as the label information for their corresponding documents in other specific domains.

Most existing keyphrase extraction datasets and studies are based on news or scientific documents and lack datasets and research related to other domains. Therefore, the task worthy of investigation is to transfer the keyphrase extraction model from the scientific domain to other domains to build a

domain-specific keyphrase extraction model with various domain generalization strategies.

7.6 Probing Pre-trained Language Model for Keyphrase Extraction

In addition to using transformer-based pre-trained language models (e.g., BERT) in NLP tasks and end applications, research has also been done on BERT, especially to reveal what linguistic information is available in different parts of the model (Jawahar et al., 2019; de Vries et al., 2020; Chen et al., 2021). It has been noted that BERT progressively acquires linguistic information roughly in the same order as the classic language processing pipeline (Tenney et al., 2019a,b): surface features are expressed in lower layers, syntactic features more in middle layers, and semantic ones in higher layers (Jawahar et al., 2019). Making full use of the above hierarchy information may effectively improve the performance of keyphrase extraction.

8 Limitations

The main goal of this paper is to provide a survey of the existing models. Since we do not propose new models, there are no potential social risks to the best of our knowledge. Our work may benefit the research community by providing more introspection into the current state-of-the-art neural keyphrase extraction approaches with pre-trained language models.

9 Acknowledgments

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

References

Joshua Ainslie, Santiago Ontañón, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 268–284. Association for Computational Linguistics.

Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *WWW*, pages 2551–2557. ACM.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie - extracting keyphrases and relations from scientific publications. In *SemEval@ACL*, pages 546–555. Association for Computational Linguistics.

Marco Basaldella, Elisa Antolli, Giuseppe Serra, and Carlo Tasso. 2018. Bidirectional LSTM recurrent neural network for keyphrase extraction. In *Digital Libraries and Multimedia Archives - 14th Italian Research Conference on Digital Libraries, IRCDL 2018, Udine, Italy, January 25-26, 2018, Proceedings*, volume 806 of *Communications in Computer and Information Science*, pages 180–187. Springer.

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

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

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2016. Keyphrase annotation with graph co-ranking. In *COLING*, pages 2945–2955. ACL.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018a. A text feature based automatic keyword extraction method for single documents. In *ECIR*, volume 10772 of *Lecture Notes in Computer Science*, pages 684–691. Springer.

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

Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 2014. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *EMNLP*, pages 1435–1446. ACL.

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

Soheil Danesh, Tamara Sumner, and James H. Martin. 2015. Sgrank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. In **SEM@NAACL-HLT*, pages 117–126. The *SEM 2015 Organizing Committee.

Wietse de Vries, Andreas van Cranenburgh, and Malvina Nissim. 2020. What's so special about bert's layers? a closer look at the nlp pipeline in monolingual and multilingual models. In *EMNLP (Findings)*, pages 4339–4350. Association for Computational Linguistics.

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

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

Samhaa R. El-Beltagy and Ahmed A. Rafea. 2009. Kp-miner: A keyphrase extraction system for english and arabic documents. *Inf. Syst.*, 34(1):132–144.

Corina Florescu and Cornelia Caragea. 2017a. A new scheme for scoring phrases in unsupervised keyphrase extraction. In *ECIR*, volume 10193 of *Lecture Notes in Computer Science*, pages 477–483.

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

Sujatha Das Gollapalli, Xiaoli Li, and Peng Yang. 2017. Incorporating expert knowledge into keyphrase extraction. In *AAAI*, pages 3180–3187. AAAI Press.

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

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

Chong Huang, Yonghong Tian, Zhi Zhou, Charles X. Ling, and Tiejun Huang. 2006. Keyphrase extraction using semantic networks structure analysis. In *ICDM*, pages 275–284. IEEE Computer Society.

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

Anette Hulth. 2004. Enhancing linguistically oriented automatic keyword extraction. In *HLT-NAACL (Short Papers)*. The Association for Computational Linguistics.

James Jardine and Simone Teufel. 2014. Topical PageRank: A model of scientific expertise for bibliographic search. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 501–510, Gothenburg, Sweden. Association for Computational Linguistics.

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

Xin Jiang, Yunhua Hu, and Hang Li. 2009. A ranking approach to keyphrase extraction. In *SIGIR*, pages 756–757. ACM.

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

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

M. Krapivin and M. Marchese. 2009. Large dataset for keyphrase extraction.

Mayank Kulkarni, Debanjan Mahata, Ravneet Arora, and Rajarshi Bhowmik. 2022. Learning rich representation of keyphrases from text. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 891–906. Association for Computational Linguistics.

Niraj Kumar and Kannan Srinathan. 2008. Automatic keyphrase extraction from scientific documents using n-gram filtration technique. In *ACM Symposium on Document Engineering*, pages 199–208. ACM.

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

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *EMNLP/IJCNLP (1)*, pages 3728–3738. Association for Computational Linguistics.

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

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *EMNLP*, pages 257–266. ACL.

Debanjan Mahata, John Kuriakose, Rajiv Ratn Shah, and Roger Zimmermann. 2018a. Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 634–639. Association for Computational Linguistics.

Debanjan Mahata, Rajiv Ratn Shah, John Kuriakose, Roger Zimmermann, and John R. Talburt. 2018b. Theme-weighted ranking of keywords from text documents using phrase embeddings. In *IEEE 1st Conference on Multimedia Information Processing and Retrieval, MIPR 2018, Miami, FL, USA, April 10-12, 2018*, pages 184–189. IEEE.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *ACL*, pages 582–592. Association for Computational Linguistics.

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

Funan Mu, Zhenting Yu, Lifeng Wang, Yequan Wang, Qingyu Yin, Yibo Sun, Liqun Liu, Teng Ma, Jing Tang, and Xing Zhou. 2020. Keyphrase extraction with span-based feature representations. *CoRR*, abs/2002.05407.

David Newman, Nagendra Koilada, Jey Han Lau, and Timothy Baldwin. 2012. Bayesian text segmentation for index term identification and keyphrase extraction. In *COLING*, pages 2077–2092. Indian Institute of Technology Bombay.

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

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *ICADL*, volume 4822 of *Lecture Notes in Computer Science*, pages 317–326. Springer.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2018. Local word vectors guiding keyphrase extraction. *Inf. Process. Manag.*, 54(6):888–902.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2019. A review of keyphrase extraction. *CoRR*, abs/1905.05044.

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

Dhruva Sahrawat, Debanjan Mahata, Haimin Zhang, Mayank Kulkarni, Agniv Sharma, Rakesh Gosangi, Amanda Stent, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2020. Keyphrase extraction as sequence labeling using contextualized embeddings. In *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part II*, volume 12036 of *Lecture Notes in Computer Science*, pages 328–335. Springer.

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

Mingyang Song, Yi Feng, and Liping Jing. 2022a. Hyperbolic relevance matching for neural keyphrase extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 5710–5720. Association for Computational Linguistics.

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

Mingyang Song, Liping Jing, and Lin Xiao. 2021. Importance Estimation from Multiple Perspectives for Keyphrase Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Si Sun, Chenyan Xiong, Zhenghao Liu, Zhiyuan Liu, and Jie Bao. 2020a. Joint keyphrase chunking and salience ranking with bert. *CoRR*, abs/2004.13639.

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

Zhiqing Sun, Jian Tang, Pan Du, Zhi-Hong Deng, and Jian-Yun Nie. 2019. Divgraphpointer: A graph pointer network for extracting diverse keyphrases. In *SIGIR*, pages 755–764. ACM.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. Bert rediscovers the classical nlp pipeline. In *ACL (1)*, pages 4593–4601. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you

learn from context? probing for sentence structure in contextualized word representations. *CoRR*, abs/1905.06316.

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

Peter D. Turney. 1999. Learning to extract keyphrases from text. *National Research Council Canada, Institute for Information Technology, Technical Report ERB-1057*.

Peter D. Turney. 2000. Learning algorithms for keyphrase extraction. *Inf. Retr.*, 2(4):303–336.

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

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

Rui Wang, Wei Liu, and Chris McDonald. 2015. Using word embeddings to enhance keyword identification for scientific publications. In *Databases Theory and Applications - 26th Australasian Database Conference, ADC 2015, Melbourne, VIC, Australia, June 4-7, 2015. Proceedings*, volume 9093 of *Lecture Notes in Computer Science*, pages 257–268. Springer.

Yanan Wang, Qi Liu, Chuan Qin, Tong Xu, Yijun Wang, Enhong Chen, and Hui Xiong. 2018. Exploiting topic-based adversarial neural network for cross-domain keyphrase extraction. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pages 597–606. IEEE Computer Society.

Yansen Wang, Zhen Fan, and Carolyn Penstein Rosé. 2020. Incorporating multimodal information in open-domain web keyphrase extraction. In *EMNLP (1)*, pages 1790–1800. Association for Computational Linguistics.

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

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. Open domain web keyphrase extraction beyond language modeling. In *EMNLP/IJCNLP (1)*, pages 5174–5183. Association for Computational Linguistics.

Wei You, Dominique Fontaine, and Jean-Paul A. Barthès. 2009. Automatic keyphrase extraction with a refined candidate set. In *Web Intelligence*, pages 576–579. IEEE Computer Society.

Linhan Zhang, Qian Chen, Wen Wang, Chong Deng, Shiliang Zhang, Bing Li, Wei Wang, and Xin Cao. 2021. Mderank: A masked document embedding rank approach for unsupervised keyphrase extraction. *CoRR*, abs/2110.06651.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: document level pre-training of hierarchical bidirectional transformers for document summarization. *CoRR*, abs/1905.06566.

Yongzheng Zhang, A. Nur Zincir-Heywood, and Evangelos E. Milios. 2004. World wide web site summarization. *Web Intell. Agent Syst.*, 2(1):39–53.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *ACL*, pages 6197–6208. Association for Computational Linguistics.

Xuan Zhou, Xiao Zhang, Chenyang Tao, Junya Chen, Bing Xu, Wei Wang, and Jing Xiao. 2021. Multi-grained knowledge distillation for named entity recognition. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5704–5716. Association for Computational Linguistics.