On Leveraging Encoder-only Pre-trained Language Models for Effective Keyphrase Generation

Di Wu, Wasi Uddin Ahmad, Kai-Wei Chang
Department of Computer Science
University of California, Los Angeles
{diwu, kwchang}@cs.ucla.edu, wasiahmad@ucla.edu

Abstract

This study addresses the application of encoder-only Pre-trained Language Models (PLMs) in keyphrase generation (KPG) amidst the broader availability of domain-tailored encoder-only models compared to encoder-decoder models. We investigate three core inquiries: (1) the efficacy of encoder-only PLMs in KPG, (2) optimal architectural decisions for employing encoder-only PLMs in KPG, and (3) a performance comparison between in-domain encoder-only and encoder-decoder PLMs across varied resource settings. Our findings, derived from extensive experimentation in two domains reveal that with encoder-only PLMs, although keyphrase extraction with Conditional Random Fields slightly excels in identifying present keyphrases, the KPG formulation renders a broader spectrum of keyphrase predictions. Additionally, prefix-LM fine-tuning of encoder-only PLMs emerges as a strong and data-efficient strategy for KPG, outperforming general-domain seq2seq PLMs. We also identify a favorable parameter allocation towards model depth rather than width when employing encoder-decoder architectures initialized with encoder-only PLMs. The study sheds light on the potential of utilizing encoder-only PLMs for advancing KPG systems and provides a groundwork for future KPG methods. Our code and pre-trained checkpoints are released at `https://github.com/uclanlp/DeepKPG`.

Keywords: Keyphrase Extraction, Keyphrase Generation, Pre-trained Language Models, Encoder-only PLMs

1. Introduction

Keyphrases are phrases that condense salient information of a document. Given their capability to capture rich information, researchers have adopted keyphrases in various applications such as document indexing, information linking, and recommendation systems (Wu and Bolivar, 2008; Dave and Varma, 2010). Additionally, keyphrases have manifested as important attributes in information retrieval (Jones and Staveley, 1999; Song et al., 2006; Kim et al., 2013; Tang et al., 2017; Boudin et al., 2020), text summarization (Zhang et al., 2004), document clustering (Hammouda et al., 2005), and text categorization (Hulth and Megyesi, 2006; Wilson et al., 2005; Berend, 2011).

Traditionally, keyphrases are classified into two categories. A *present keyphrase* is one that is directly extractable from the document, while an *absent keyphrase* does not have a direct representation within the text (Figure 1). The task of *keyphrase extraction* (KPE) requires models to identify present keyphrases. Meng et al. (2017) extended this by introducing *keyphrase generation* (KPG), aiming to predict both present and absent keyphrases.

In recent years, the emergence of pre-trained language models (PLMs) has revolutionized the KPE and KPG landscapes, leading to their application in unsupervised KPE (Sun et al., 2020; Liang et al., 2021), sequence-labeled KPE (Sahrawat et al., 2019; Dascalu and Trăuşan-Matu, 2021), and sequence-to-sequence KPG (Liu et al., 2020, 2021a; Chowdhury et al., 2022). Remarkably, PLM-based KPG methodologies have exhibited promising performance, especially in scenarios such as zero-shot (Kulkarni et al., 2022), multilingual (Gao et al., 2022), low-resource (Wu et al., 2022), and cross-domain (Meng et al., 2023) settings.

Document title
J.F.K. Workers Moved Drugs, Authorities Say
Document body
Airline employees exploited weaknesses in security procedures to help a New York drug ring smuggle heroin and cocaine through Kennedy International Airport, federal authorities charged yesterday. ...
Present and Absent Keyphrases
smuggling, heroin, kennedy international airport, drug abuse and traffic, crime and criminals, cocaine and crack cocaine

Figure 1: An example of a news article with its present and absent keyphrases highlighted in blue and red, respectively.

However, the deployment of PLM-driven systems for keyphrase prediction in practical applications presents a dilemma. While the ability of KPG models to generate absent keyphrases is lauded, particularly due to its impact in retrieval systems (Boudin and Gallina, 2021), there is a nuanced inclination towards KPE techniques for domain-specific applications. Such techniques predominantly harness in-domain encoder-only PLMs like BERT (Devlin et al., 2019; Gururangan et al., 2020), as opposed to the KPG methods which often employ prefix-LMs (Wu et al., 2021) or sequence-to-sequence PLMs (Zhao et al., 2022). The abundance of encoder-only models across specific domains accentuates

LREC-COLING 2024, pages 12370–12384
20-25 May, 2024.

this preference (Figure 2), while domain-oriented encoder-decoder models remain relatively scarce. Motivate by this challenge, this paper studies the following research questions:

- Can we use encoder-only PLMs for KPG and achieve a similar present keyphrase performance as using them for KPE?
- What is the best architectural choice for using encoder-only PLMs for KPG?
- How do in-domain encoder-only PLMs compare to encoder-decoder PLMs in rich and low resource settings?

To answer these questions, we investigate four formulations for KPE and KPG with encoder-only PLMs. For KPE, we fine-tune on sequence labeling with or without Conditional Random Field (Lafferty et al., 2001). For KPG, we fine-tune either using prefix-LM style attention masks (Dong et al., 2019), or initialize an encoder-decoder architecture with encoder-only PLMs (Rothe et al., 2020). With extensive experiments on datasets covering the science and news domain, our main findings are summarized as follows.

1. For present keyphrases, KPE with Conditional Random Fields (CRF) exhibits a slight advantage over KPG in terms of macro F1@M. However, training encoder-only models for KPG produce much more keyphrase predictions including both present and absent keyphrases.
2. Prefix-LM fine-tuning of encoder-only PLMs is a strong and data-efficient KPG method, even outperforming general-domain seq2seq PLMs of the same size.
3. For encoder-decoder architecture initialized with encoder-only PLMs, how parameters are allocated strongly affects the performance. Specifically, model *depth* should be prioritized over the width, and a *deep encoder with a shallow decoder* outperforms the reverse for keyphrase quality and inference latency.
4. We conducted an evaluation of applying the encoder-only SciBERT and NewsBERT for KPG inside and outside of their pre-training domain. Compared to the general-domain BERT, both models demonstrate better in-domain performance. However, SciBERT transfers well to the news domain while NewsBERT cannot transfer to the science domain.

We hope this empirical study can shed light on the opportunities of leveraging encoder-only PLMs for building KPG systems, facilitating the development of more effective approaches. Our code and the pre-trained NewBERT model will be released at `https://github.com/uclanlp/DeepKPG`.

Model Name	Reference	Domain
SciBERT	Beltagy et al. (2019)	Science
BioBERT	Lee et al. (2019)	Science
ChemBERTa	Chithrananda et al. (2020)	Science
Bio_RoBERTa	Gururangan et al. (2020)	Science
CS_RoBERTa	Gururangan et al. (2020)	Science
BioMegatron	Shin et al. (2020)	Science
MatSciBERT	Gupta et al. (2021)	Science
PubMedBERT	Gu et al. (2022)	Science
MatBERT	Trewartha et al. (2022)	Science
BatteryBERT	Huang and Cole (2022)	Science
ClinicalBERT	Alsentzer et al. (2019)	Social
FinBERT	Liu et al. (2021b)	Social
LEGAL-BERT	Chalkidis et al. (2020)	Social
JobBERT	Zhang et al. (2022)	Social
PrivBERT	Srinath et al. (2021)	Social
SportsBERT	Srinivasan and Mashetty	Social
Twitter-roberta	Barbieri et al. (2020)	Web
BERTweet	Nguyen et al. (2020)	Web
News_RoBERTa	Gururangan et al. (2020)	Web
NewsBERT	this work	Web
Reviews_RoBERTa	Gururangan et al. (2020)	Web
HateBERT	Caselli et al. (2021)	Web

Figure 2: Domain-specific encoder-only PLMs are available in a variety of domains. No prior work considered using these "domain experts" for KPG. In this paper, we show that these specialized encoder-only PLMs can be used to build strong and resource-efficient KPG models.

2. Related Work

Keyphrase Extraction Early work on KPE mainly followed a pipelined approach. First, keyphrase candidates (usually noun phrases) are extracted by rules such as regular expression matching on part-of-speech tags. Then, various scoring methods are used to rank the candidates. The ones with the highest scores are returned as keyphrase predictions (Hulth, 2003a; Mihalcea and Tarau, 2004; Wan and Xiao, 2008b; Bougouin et al., 2013; Sun et al., 2020; Boudin, 2018; Liang et al., 2021). In terms of supervised approaches, one line of research adopts the sequence labeling formulation, removing the need for selecting candidates (Zhang et al., 2016; Luan et al., 2017; Sahrawat et al., 2019). Others instead aim to learn a more capable ranking function with pre-trained language models (Song et al., 2021, 2022).

Keyphrase Generation Meng et al. (2017) propose the task of Deep Keyphrase Generation and a strong baseline model CopyRNN. Later works improve the architecture by adding correlation constraints (Chen et al., 2018) and linguistic constraints (Zhao and Zhang, 2019), exploiting learning signals from titles (Ye and Wang, 2018; Chen et al., 2019b), and hierarchical modeling the phrases and words (Chen et al., 2020). Yuan et al. (2020) reformulate the problem as generating a sequence of keyphrases, while Ye et al. (2021) further introduces a set generation reformulation to remove the influence of difference target phrase ordering.

Other works include incorporating reinforcement learning (Chan et al., 2019; Luo et al., 2021), GANs (Swaminathan et al., 2020), and unifying KPE with KPG (Chen et al., 2019a; Ahmad et al., 2021; Wu et al., 2021). Meng et al. (2021) conduct an empirical study on architecture, generalizability, phrase order, and decoding strategies, mainly focusing on RNNs and Transformers trained from scratch.

KPE and KPG with PLMs More recently, Sahrawat et al. (2019), Liu et al. (2020), Liu et al. (2021a), and Dascalu and Trăuşan-Matu (2021) have considered using pre-trained BERT (Devlin et al., 2019) for keyphrase extraction and generation. Wu et al. (2021) propose to fine-tune a prefix-LM to jointly extract and generate keyphrases. In addition, Chowdhury et al. (2022), Kulkarni et al. (2022), Wu et al. (2022), and Gao et al. (2022) use seq2seq PLMs such as BART or T5 in their approach. Kulkarni et al. (2022) use keyphrase generation as a pre-training task to learn strong PLM-based representations. Wu et al. (2023) investigate why and how encoder-decoder PLMs can be effective for KPG, focusing on model selection and decoding strategies. In this paper, we aim to bridge the gap in understanding how encoder-only PLMs could be best utilized for KPG.

3. Evaluation Setup

In this section, we formulate the KPE and KPG tasks and introduce the evaluation setup. We end by discussing the PLMs we are considering.

3.1. Problem Definition

We view a keyphrase example as a triple $(\mathbf{x}, \mathbf{y_p}, \mathbf{y_a})$, corresponding to the input document $\mathbf{x} = (x_1, x_2, ..., x_d)$, the set of present keyphrases $\mathbf{y_p} = \{y_{p_1}, y_{p_2}, ..., y_{p_m}\}$, and the set of absent keyphrases $\mathbf{y_a} = \{y_{a_1}, y_{a_2}, ..., y_{a_n}\}$. For both keyphrase extraction and generation, $\mathbf{x}$ consists of the title and the document body, concatenated with a special $[sep]$ token. Following Meng et al. (2017), each $y_{p_i} \in \mathbf{y_p}$ is a subsequence of $\mathbf{x}$, and each $y_{a_i} \in \mathbf{y_a}$ is not a subsequence of $\mathbf{x}$.

Using this formulation, the **keyphrase extraction** (KPE) task requires the model to predict $\mathbf{y_p}$. On the other hand, the **keyphrase generation** (KPG) task requires the prediction of $\mathbf{y_p} \cup \mathbf{y_a}$.

3.2. Benchmarks

We evaluate using two widely-used KPG benchmarks in the science and the news domains. Table 1 summarizes the statistics of the testing datasets.

SciKP Meng et al. (2017) introduce KP20k, which contains 500k Computer Science papers. Following their work, we train on KP20k and evaluate on the KP20k test set.

Dataset	#Examples	#KP	%AKP	|KP|
KP20k	20000	5.3	37.1	2.0
KPTimes	20000	5.0	37.8	2.0

Table 1: Test sets statistics. #KP, %AKP, and |KP| refers to the average number of keyphrases per document, the percentage of absent keyphrases, and the average number of words in each keyphrase.

KPTimes Introduced by Gallina et al. (2019), KPTimes is a keyphrase dataset in the news domain containing over 250k examples. We train on the KPTimes train set and report the performance on the union of the KPTimes test set and the out-of-distribution test set JPTimes.

3.3. Evaluation

Each method's predictions are normalized into a sequence of present and absent keyphrases. The phrases are ordered by the position in the source document for the sequence labeling approaches to obtain the keyphrase predictions. Then, we apply the Porter Stemmer (Porter, 1980) to the output and target phrases and remove the duplicated phrases from the output. Following Chan et al. (2019) and Chen et al. (2020), we separately report the macro-averaged F1@5 and F1@M scores for present and absent keyphrases. For all the results except the ablation studies, we train with three different random seeds and report the averaged scores.

3.4. Baselines

We consider four supervised KPG baselines.
CatSeq (Yuan et al., 2020) is a CopyRNN (Meng et al., 2017) trained on generating keyphrases as a sequence, separated by the separator token.
ExHiRD-h (Chen et al., 2020) improves CatSeq with a hierarchical decoding framework and a hard exclusion mechanism to reduce duplicates.
Transformer (Ye et al., 2021) is the self-attention based seq2seq model (Vaswani et al., 2017) with copy mechanism (See et al., 2017).
SetTrans (Ye et al., 2021) performs order-agnostic KPG. The model uses control codes trained via a k-step target assignment algorithm.
For KPE, we also include a range of ranking-based unsupervised KPE methods and a **Transformer** as baseline references. As the main focus of the paper is studying encoder-only PLMs, we only compare with the strongest non-PLM baselines.

3.5. Considered PLMs

Encoder-Only PLMs We study **BERT** (Devlin et al., 2019), **SciBERT** (Beltagy et al., 2019), **NewsBERT** (details below), and five pre-trained BERT checkpoints from Turc et al. (2019) with the hidden size 768, 12 attention heads per layer, and 2, 4, 6, 8, and 10 layers.

Encoder-Decoder PLMs As this paper focuses on encoder-only PLMs, we only include **BART** (Lewis et al., 2020) and its in-domain variations **SciBART** (Wu et al., 2023) and **NewsBART** (details below) as references.

Domain-Specific PLMs Inspired by Wu et al. (2023), we pre-train **NewsBART** using the RealNews dataset (Zellers et al., 2019), which contains around 130GB of news text from 2016 to 2019. We follow Wu et al. (2023) for data cleaning and preprocessing. For training, we mask 30% of tokens and sample the spans from a Poisson distribution ($\lambda = 3.5$). For 10% of the masking spans selected to mask, we replace them with a random token instead of the special `<mask>` token. We set the maximum sequence length to 512. Starting from BART-base, the model is trained for 250k steps using the Adam optimizer with batch size 2048, learning rate 3e-4, 10k warm-up steps, and polynomial learning rate decay. The pre-training was conducted on 8 Nvidia A100 GPUs and costed approximately 5 days.

We also pre-train a **NewsBERT** model using the same data on masked language modeling with 15% dynamic masking. Starting from BERT-base (Devlin et al., 2019), the model is trained for 250k steps with batch size 512, learning rate 1e-4, 5k warm-up steps, and linear learning rate decay. We use the Adam optimizer for pre-training. The pre-training was conducted on 8 Nvidia V100 GPUs and costed approximately 7 days.

4. Modeling Approaches

Next, we introduce the modeling methods we adopt in this paper for KPE and KPG.

4.1. Keyphrase Extraction (KPE)

For KPE, we use a sequence labeling formulation. Each $x_i \in \mathbf{x}$ is assigned a label $c_i \in \{B_{kp}, I_{kp}, O\}$ depending on x being the beginning token of a present keyphrase, the subsequent token of a present keyphrase, or otherwise. The model is required to predict the label for each token.

In this paper, we fine-tune three encoder-only PLMs: **BERT** (Devlin et al., 2019), **SciBERT** (Beltagy et al., 2019), and **NewsBERT** (section 3.5) [1]. We add a fully connected layer for each model to project the hidden representation of every token into three logits representing B_{kp}, I_{kp}, and O. The model is trained on the cross-entropy loss. We also experiment with using Conditional Random Field (Lafferty et al., 2001) to model the sequence-level transitions. We use **+CRF** to refer to this setting.

4.2. Keyphrase Generation (KPG)

Following Yuan et al. (2020), we use a special separator token `;` to join all the keyphrases in a sequence $\mathbf{y} = (y_{p_1}\ ;\ ...\ ;\ y_{p_m}\ ;\ y_{a_1}\ ;\ ...\ ;\ y_{a_m})$. Using this sequence generation formulation, we fine-tune the encoder-decoder PLMs for KPG. The models are trained with the cross-entropy loss for generating the target sequence $\mathbf{y}$. Next, we discuss the methods we explore for training encoder-only PLMs on KPG.

BERT2BERT We construct seq2seq keyphrase generation models by separately initializing the encoder and the decoder with encoder-only PLMs. Following Rothe et al. (2020), we add cross-attention layers to the decoder. We use five pre-trained BERT checkpoints from Turc et al. (2019) with hidden size 768, 12 attention heads per layer, and 2, 4, 6, 8, and 10 layers. **B2B-**e**+**d denotes a BERT2BERT model with an e-layer pre-trained BERT as the encoder and a d-layer pre-trained BERT as the decoder. We use BERT2RND (**B2R**) to denote randomly initializing the decoder and RND2BERT (**R2B**) to denote randomly initializing the encoder. The models are fine-tuned using the same sequence generation formulation.

Prefix-LM Dong et al. (2019) propose jointly pre-training for unidirectional, bidirectional, and seq2seq language modeling by controlling attention mask patterns. In the seq2seq setup, the input is $\mathbf{x}\ [eos]\ \mathbf{y}$. The attention mask is designed such that tokens in $\mathbf{x}$ are only allowed to attend to $\mathbf{x}$, and that tokens in $\mathbf{y}$ are allowed to attend to tokens on their left. Using this formulation, we fine-tune encoder-only PLMs for seq2seq keyphrase generation. Following Dong et al. (2019), we mask and randomly replace tokens from $\mathbf{y}$ and train the model on the cross-entropy loss between its reconstruction and the original sequence. We call our models **BERT-G**, **SciBERT-G**, and **NewsBERT-G**. Our approach is different from Wu et al. (2021) in that (1) we use encoder-only PLMs that are not pre-trained on prefix-LM and (2) we consider letting the model directly generate the present keyphrases instead of adding new layers dedicated to KPE.

4.3. Implementation Details

Keyphrase Extraction We implemented our models with Huggingface Transformers[2] and TorchCRF[3]. The models are trained for ten epochs with early stopping. We use a learning rate of 1e-5 with linear decay and batch size 32 for most models (see appendix for all the hyperparameters). We use AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

[1] In this study, we use the base variants of all the encoder-only models unless otherwise specified.

[2] https://github.com/huggingface/transformers

[3] https://github.com/s14t284/TorchCRF

Method	|M|	KP20k			KPTimes		
		#KP	F1@5	F1@M	#KP	F1@5	F1@M
Keyphrase extraction							
BERT	110M	(3.4, 0.0)	27.9	38.9	(2.5, 0.0)	34.0	49.3
SciBERT	110M	(3.1, 0.0)	28.6	40.5	(2.2, 0.0)	31.8	47.7
NewsBERT	110M	(3.0, 0.0)	25.8	37.5	(2.5, 0.0)	34.5	50.4
BERT+CRF	110M	(3.2, 0.0)	28.0	40.6	(2.4, 0.0)	33.9	49.9
SciBERT+CRF	110M	(2.8, 0.0)	28.4	**42.1**	(2.1, 0.0)	31.8	48.1
NewsBERT+CRF	110M	(2.8, 0.0)	26.8	39.7	(2.4, 0.0)	**34.9**	**50.8**
Present keyphrase generation							
BERT-G	110M	(4.1, 0.9)	31.3	37.9	(2.3, 2.2)	32.3	47.4
SciBERT-G	110M	(4.3, 1.1)	**32.8**	39.7	(2.4, 2.0)	33.0	48.4
NewsBERT-G	110M	(4.2, 0.7)	29.9	36.8	(2.4, 2.1)	33.0	48.0

Table 2: Present keyphrase performance of encoder-only PLM-based sequence labeling and sequence generation approaches. #KP = average number of (present, absent) keyphrase predictions per document. The best results are boldfaced. We find that the prefix-LM approach allows the encoder-only models to generate much more keyphrases without greatly sacrificing F1@M.

Keyphrase Generation For BART and BERT2BERT, we use Huggingface Transformers to implement the fine-tuning. To fine-tune SciBART-base, SciBART-large, and NewsBART-base, we use the fairseq[4]. For BERT-G, SciBERT-G, NewsBART-G, and UniLM, we implement the training based on Dong et al. (2019)'s original implementations [5]. We set the maximum source and target length to 464 tokens and 48 tokens, respectively. We mask 80% of the target tokens and randomly replace an additional 10%. We use the AdamW optimizer.

Hyperparameters For each of the PLM-based KPE and KPG methods, we perform a careful hyper-parameter search over the learning rate, learning rate schedule, batch size, and warm-up steps. The corresponding search spaces are {1e-5, 5e-4}, {linear, polynomial}, {16, 32, 64, 128}, and {500, 1000, 2000, 4000}. For all models, we use the Adaom optimizer, apply a linear learning rate schedule, and do early stopping. The best hyperparameters found are presented in Table 6.

The fine-tuning experiments are run on a local GPU server with Nvidia GTX 1080 Ti and RTX 2080 Ti GPUs. We use at most 4 GPUs and gradient accumulation to achieve the desired batch sizes. We use greedy decoding for all the models. For evaluation, we follow Chan et al. (2019)'s implementation.

5. Results

We aim to address the following questions.

1. Is the KPG formulation less suitable to encoder-only PLMs compared to KPE?
2. Can encoder-only PLMs generate better keyphrases than encoder-decoder PLMs?
3. What is the best parameter allocation strategy for using encoder-decoder PLMs to balance the performance and computational cost?

5.1. Extraction vs. Generation for encoder-only PLMs

To begin with, we investigate the viability of utilizing encoder-only PLMs for KPG by comparing three formulations for present keyphrases: (1) sequence labeling via token-wise classification, (2) sequence labeling with CRF, and (3) sequence generation. The results are presented in Table 2.

For all the models studied, we find that adding a CRF layer consistently improves the KPE performance. On KP20k, we find that this sequence labeling objective can guide the generation of more accurate (reflected by high F1@M) but much fewer keyphrases, leading to a significantly lower F1@5. On the other hand, the prefix-LM approach allows generating more present keyphrases as well as predicting absent keyphrases, resulting in a much higher F1@5 and a comparable F1@M. On KPTimes, we find sequence labeling achieving better accuracy for BERT and NewsBERT, while prefix-LM achieves a better accuracy for SciBERT.

To summarize, an encoder-only PLM can produce vastly different behaviors depending on the formulation for KPE and KPG. The prefix-LM approach mostly matches the present keyphrase performance of sequence labeling methods and promises to generate many more keyphrases including absent ones. Next, we focus on the two formulations for KPG using encoder-only PLMs and discuss their performance in the context of encoder-decoder PLMs and non-PLM KPG models.

[4] `https://github.com/facebookresearch/fairseq`

[5] `https://github.com/microsoft/unilm`

Method	|M|	KP20k Present F1@5	KP20k Present F1@M	KP20k Absent F1@5	KP20k Absent F1@M	KPTimes Present F1@5	KPTimes Present F1@M	KPTimes Absent F1@5	KPTimes Absent F1@M
Encoder-only PLMs									
BERT-G	110M	31.3	37.9	1.9	3.7	32.3	47.4	16.5	24.6
B2B-8+4	143M	32.2	38.0	2.2	4.2	33.8	48.6	16.8	24.5
SciBERT-G	110M	32.8	**39.7**	2.4	4.6	33.0	48.4	15.7	24.7
NewsBERT-G	110M	29.9	36.8	1.3	2.6	33.0	48.0	17.0	25.6
Encoder-Decoder PLMs									
BART	140M	32.2	38.8	2.2	4.2	**35.9**	**49.9**	17.1	24.9
SciBART	124M	34.1	39.6	2.9	5.2	34.8	48.8	17.2	24.6
NewsBART	140M	32.4	38.7	2.2	4.4	35.4	49.8	17.6	**26.1**
Non-PLM baselines									
CatSeq	21M	29.1	36.7	1.5	3.2	29.5	45.3	15.7	22.7
ExHiRD-h	22M	31.1	37.4	1.6	2.5	32.1	45.2	13.4	16.5
Transformer	98M	33.3	37.6	2.2	4.6	30.2	45.3	17.1	23.1
SetTrans	98M	**35.6**	39.1	**3.5**	**5.8**	35.6	46.3	**19.8**	21.9

Table 3: A comparison across encoder-only and encoder-decoder PLMs from different domains for keyphrase generation. The best results are boldfaced, and the best encoder-only PLM results are underlined. We use different colors for PLMs in the science and the news domain.

5.2. Can encoder-only PLMs generate better keyphrases than BART and strong non-PLM methods?

In this section, we compare (1) KPG via prefix-LM, (2) KPG with the best performance of BERT2BERT models with a 12-layer budget (full details in §5.3), (3) BART-based seq2seq PLMs, and (4) four strong non-PLM KPG models. Table 3 presents the major results on KP20k and KPTimes.

BERT vs. BART We start with the surprising result that strong KPG models can be obtained through prefix-LM fine-tuning of encoder-only PLMs. The strength is especially notable with *in-domain* BERT models. On KP20k, SciBERT-G outperforms BART-base on all the metrics. On KPTimes, NewsBERT-G has a comparable F1@5 and better F1@M for absent KPG compared to the BART-base. We also provide the results from in-domain BART models as they have been shown to achieve strong performance gains (Wu et al., 2023). We note that such models are often not as accessible as the in-domain BERT model in highly specialized domains like KP20k. Our findings highlight the viability of directly using the in-domain encoder-only PLMs for KPG and obtain comparable results. The trend is more evident in the low-resource scenario. As shown in Figure 3, SciBERT-G is as resource-efficient as SciBART, outperforming BART in every low-resource setting on KP20k.

BERT vs. non-PLM baselines Given a large training set (500k for KP20k and 250k for KPTimes), the best-performing encoder-only KPG methods outperform SetTrans on KPTimes but only achieve the same or worse performance compared to SetTrans on KP20k. This is expected since the non-

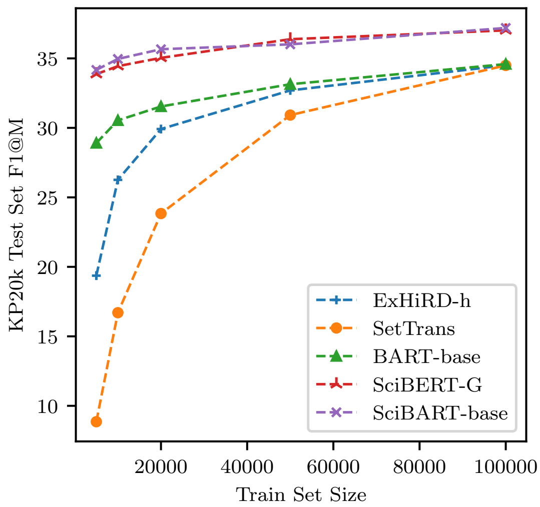

Figure 3: Present keyphrase generation performance of different methods as a function of training set size. Fine-tuning in-domain PLMs is much more data-efficient than the other approaches.

PLM architectures introduce task-specific designs to enable more effective training or inference. However, in the low-resource scenario, we find that the preference for encoder-only PLMs is clear. As shown in Figure 3, SciBERT-G only requires 5k data to achieve the same F1@M of SetTrans fine-tuned with 100k data.

Prefix-LM vs. BERT2BERT We observe that combining two smaller-sized BERT models and training on KPG outperform BERT-G despite a similar model size. On KPTimes, the B2B model with an 8-layer encoder and a 4-layer decoder achieves the best present keyphrase performance among all encoder-only PLMs. The model also has **a lower inference latency** due to its shallow decoder (further explored in §5.3). However, this formulation may not be as resource-efficient as prefix-LM, as it randomly initializes cross-attention weights.

To summarize, the two KPG formulations we explore enable encoder-only PLMs to outperform general-domain seq2seq PLMs on KPG, with potentially better data efficiency (when in-domain models are employed) and compute efficiency (when BERT2BERT formulation is used). The results suggest a new possibility of utilizing the more accessible encoder-only models to build stronger keyphrase generation systems, especially in highly specialized domains.

5.3. What is the best parameter allocation for BERT2BERT?

Observing that the BERT2BERT setup produces strong keyphrase generation models, we further investigate the optimal parameter allocation strategy. Specifically, under a given parameter budget, should depth (i.e., more layers) or width (fewer layers, more parameters per layer) be prioritized? Moreover, should the encoder or the decoder be allocated more parameters? As computational resources prevent us from extensive pre-training, we use the BERT2BERT setting with different sized BERT models instead.

Depth should be prioritized over width. We design four pairs of B2B models with different total parameter budgets: 20M, 50M, 85M, and 100M. Each pair contains (a) a model that prioritizes depth and (b) a model that prioritizes width. We ensure that (a) and (b) have similar encoder depth to decoder depth ratios (except the 85M group). The results on KP20k are presented in Table 4. For all the groups, model (a) performs significantly better despite having slightly fewer parameters.

A deep encoder with a shallow decoder is preferred. We fix a budget of 12 layers and experiment with five encoder-decoder combinations. Table 5 presents the results on KP20k and KPTimes. For both datasets, we find that the performance increases sharply and then plateaus as the depth of the encoder increases. With the same budget, **a deep encoder followed by a shallow decoder is clearly preferred over a shallow encoder followed by a deep decoder**. We hypothesize that comprehending the input article is important and challenging while generating a short string comprising several phrases based on the encoded representation does not necessarily require a PLM-based decoder.

To verify, we further experiment with randomly initializing either the encoder ("R2B") or the decoder

Model setup	|M|	Present KPs		Absent KPs	
		F1@5	F1@M	F1@5	F1@M
B2B-12+12-128	13M	**26.4**	**33.8**	**1.0**	**2.1**
B2B-2+2-256	20M	22.1	31.9	**1.0**	**2.1**
B2B-12+12-256	38M	**30.8**	**36.7**	**1.6**	**3.3**
B2B-2+2-512	47M	27.5	34.7	1.4	3.0
B2B-10+4-512	81M	**31.4**	**37.4**	**1.8**	**3.7**
B2B-2+2-768	82M	29.4	35.5	**1.8**	3.5
B2B-12+6-512	95M	**31.9**	**38.2**	**2.1**	**4.0**
B2B-4+2-768	96M	30.8	37.1	2.0	**4.0**

Table 4: Comparison between different parameter allocation strategies. The best performance of each group is boldfaced. B2B-e+d-h denotes a B2B model with e encoder layers, d encoder layers, and hidden size h.

e-d	|M|	Arch.	KP20k		KPTimes	
			F1@5	F1@M	F1@5	F1@M
Present keyphrase generation						
2-10	158M	B2B	30.4	36.4	31.6	46.5
4-8	153M	B2B	31.7	37.7	32.9	47.6
		R2B	26.3	35.2	28.2	43.3
		B2R	31.7	37.9	32.6	47.5
6-6	148M	B2B	32.1	37.7	**33.8**	48.4
		R2B	26.4	35.3	27.8	42.9
		B2R	32.0	38.4	33.3	48.2
8-4	143M	B2B	**32.2**	**38.0**	**33.8**	**48.6**
		R2B	27.3	35.4	27.8	42.8
		B2R	31.2	37.9	33.2	48.0
10-2	139M	B2B	31.7	38.0	33.5	48.4
Absent keyphrase generation						
2-10	158M	B2B	2.1	3.9	16.2	23.2
4-8	153M	B2B	**2.2**	4.1	15.9	23.6
		R2B	2.5	4.2	14.7	24.3
		B2R	2.2	4.2	16.5	24.1
6-6	148M	B2B	**2.2**	4.1	16.4	24.1
		R2B	2.6	4.3	14.5	20.8
		B2R	2.3	4.4	16.2	23.9
8-4	143M	B2B	**2.2**	**4.2**	**16.8**	**24.5**
		R2B	2.4	4.1	14.9	21.0
		B2R	2.1	4.1	16.8	24.7
10-2	139M	B2B	2.1	4.1	**16.8**	**24.5**

Table 5: A comparison between different BERT2BERT architectures and initializations. In e-d, e and d indicate the number of encoder and decoder layers, respectively. The best results among the B2B models are boldfaced.

("B2R"). The results are shown in Table 5. For both KP20k and KPTimes, we observe that randomly initializing the encoder harms the performance, while randomly initializing the decoder does not significantly impact the performance (the absent KPG is even improved in some cases).

Latency Concerns We further explore the impact on the inference speed of different BERT2BERT configurations. Specifically, we measure and compare the inference throughput of B2B-2+10, B2B-

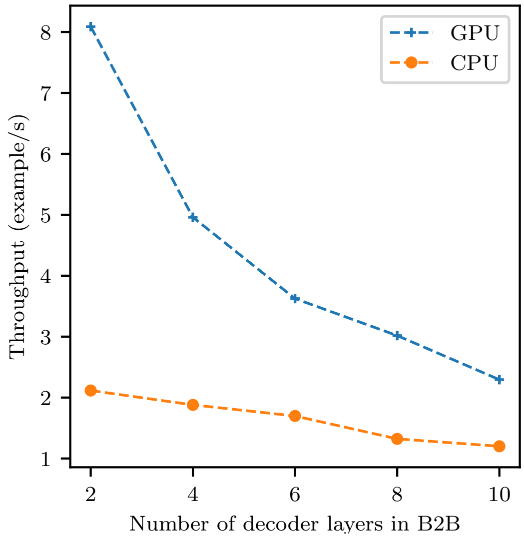

Figure 4: Inference speed of BERT2BERT models with different encoder-decoder configurations on GPU and CPU. All the data points are obtained with BERT2BERT models with 12 layers. A model with x decoder layers has 12-x encoder layers.

4+8, B2B-6+6, B2B-8+4, and B2B-10+2 in GPU and CPU environments. We use the best model trained on KP20k and test on the KP20k test set with batch size 1, no padding. We use an Nvidia GTX 1080 Ti card for GPU and test on the full KP20k test set. We use a local server with 40 cores for CPU and test on the first 1000 examples from the KP20k test set. No inference acceleration libraries are used. Figure 4 reports the averaged throughput (in example/s) across three runs. The throughput decreases significantly with deeper decoders for both CPU and GPU. B2B-8+4 achieves better performance than B2B-6+6 while being 37% faster on GPU and 11% faster on CPU. In conclusion, with a limited parameter budget, we recommend using *more layers* and *a deep-encoder with shallow-decoder* architecture.

6. Conclusion

This paper unveils the viable application of encoder-only Pre-trained Language Models (PLMs) in the Keyphrase Generation (KPG) task. We introduce two formulations for utilizing BERT-like PLMs for KPG and our systematic experiments demonstrate their strong performance, resource efficiency, and competitive inference latency. Our study opens up promising possibilities for a wider range of in-domain models of efficient and effective keyphrase generation systems.

However, the study is not without limitations. For instance, we only focus on small-sized models below 500M. In addition, the transferability of domain-specific models, as observed between SciBERT and NewsBERT, hints at a potential domain bias that warrants further examination. Future endeavors could delve into a more exhaustive evaluation across a spectrum of encoder-decoder PLMs, investigate the underlying factors affecting model transferability, and explore the augmentation of encoder-only PLMs with additional external knowledge to further enhance keyphrase prediction accuracy across diverse domains. The insights in our work lay a preliminary foundation for more comprehensive inquiries into architecturally diverse PLMs to continually refine KPG methodologies.

Model	D	B_s	E	W_s	LR
Keyphrase extraction					
Transformer	0.1	32	10	2000	3e-5
BERT	0.1	32	10	1000	1e-5
SciBERT	0.1	32	10	1000	1e-5
Transformer+CRF	0.1	32	10	2000	3e-5
BERT+CRF	0.1	32	10	2000	1e-5
SciBERT+CRF	0.1	32	10	2000	1e-5
RoBERTa+CRF	0.1	32	10	2000	1e-5
Keyphrase generation					
BERT-G	0.1	64	6	4000	1e-4
SciBERT-G	0.1	128	6	2000	1e-4
UniLM	0.1	128	6	2000	1e-4
BERT2BERT	0.0	32	20	2000	5e-5
BART-base	0.1	64	15	2000	6e-5
SciBART-base	0.1	32	10	2000	3e-5
KeyBART	0.1	64	15	2000	3e-5

Table 6: Hyperparameters for fine-tuning PLMs for keyphrase extraction and keyphrase generation on KP20k. The hyperparameters are determined using the loss on the KP20k validation dataset. We follow a similar set of hyperparameters for KPTimes. D = dropout, B_s = batch size, E = number of epochs, W_s = number of warm-up steps, LR = learning rate. We use early stopping for all the models and use the model with the lowest validation loss as the final model. For all the models, we use weight decay 0.01 and a linear LR schedule.

7. Ethics Statement

RealNews is released under Apache 2.0. We conduct text cleaning and email/URL filtering on RealNews to remove sensitive information. The SciKP and KPTimes benchmarking datasets, distributed by the original authors, are utilized as-is with no additional preprocessing performed before fine-tuning except for lower casing and tokenization. We do not re-distribute any of the pre-training and benchmarking datasets.

The computational resources required for pre-training NewsBART and NewsBERT are significant, and we estimate the total CO_2 emissions to be around 600 kg using the calculation application provided by Lacoste et al. (2019). Additionally, we estimate that all fine-tuning experiments, including hyperparameter optimization, emitted around 1500 kg CO_2. To assist the community in reducing the energy required for optimizing PLMs across various NLP applications in the scientific and news domains, we provide the hyperparameter and release our pre-trained NewsBART and NewsBERT checkpoints. We will limit the access to the models with a non-commercial license. We will also release the raw predictions of our models.

8. Bibliographical References

Wasi Ahmad, Xiao Bai, Soomin Lee, and Kai-Wei Chang. 2021. Select, extract and generate: Neural keyphrase generation with layer-wise coverage attention. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1389–1404, Online. Association for Computational Linguistics.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 221–229, Brussels, Belgium. Association for Computational Linguistics.

Gábor Berend. 2011. Opinion expression mining by exploiting keyphrase extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1162–1170, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Florian Boudin. 2016. pke: an open source python-based keyphrase extraction toolkit. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 69–73, Osaka, Japan.

Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite graphs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 667–672, New Orleans, Louisiana. Association for Computational Linguistics.

Florian Boudin and Ygor Gallina. 2021. Redefining absent keyphrases and their effect on retrieval effectiveness. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4185–4193, Online. Association for Computational Linguistics.

Florian Boudin, Ygor Gallina, and Akiko Aizawa. 2020. Keyphrase generation for scientific document retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1126, Online. Association for Computational Linguistics.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. TopicRank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551, Nagoya, Japan. Asian Federation of Natural Language Processing.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. Yake! collection-independent automatic keyword extractor. In *European Conference on Information Retrieval*, pages 806–810. Springer.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Hou Pong Chan, Wang Chen, Lu Wang, and Irwin King. 2019. Neural keyphrase generation via reinforcement learning with adaptive rewards. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2163–2174, Florence, Italy. Association for Computational Linguistics.

Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. Keyphrase generation with correlation constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4057–4066, Brussels, Belgium. Association for Computational Linguistics.

Wang Chen, Hou Pong Chan, Piji Li, Lidong Bing, and Irwin King. 2019a. An integrated approach for keyphrase generation via exploring the power of retrieval and extraction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2846–2856, Minneapolis, Minnesota. Association for Computational Linguistics.

Wang Chen, Hou Pong Chan, Piji Li, and Irwin King. 2020. Exclusive hierarchical decoding for deep keyphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1095–1105, Online. Association for Computational Linguistics.

Wang Chen, Yifan Gao, Jiani Zhang, Irwin King, and Michael R. Lyu. 2019b. Title-guided encoding for keyphrase generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6268–6275.

Seyone Chithrananda, Gabriel Grand, and Bharath Ramsundar. 2020. Chemberta: Large-scale self-supervised pretraining for molecular property prediction.

Md Faisal Mahbub Chowdhury, Gaetano Rossiello, Michael Glass, Nandana Mihindukulasooriya, and Alfio Gliozzo. 2022. Applying a generic sequence-to-sequence model for simple and effective keyphrase generation.

Cristian Dascalu and Ştefan Trăuşan-Matu. 2021. Experiments with contextualized word embeddings for keyphrase extraction. In *2021 23rd International Conference on Control Systems and Computer Science (CSCS)*, pages 447–452.

Kushal S. Dave and Vasudeva Varma. 2010. Pattern based keyword extraction for contextual advertising. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 1885–1888, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Samhaa R. El-Beltagy and Ahmed Rafea. 2010. KP-miner: Participation in SemEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 190–193, Uppsala, Sweden. Association for Computational Linguistics.

Corina Florescu and Cornelia Caragea. 2017. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115, Vancouver, Canada. Association for Computational Linguistics.

Ygor Gallina, Florian Boudin, and Beatrice Daille. 2019. KPTimes: A large-scale dataset for keyphrase generation on news documents. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 130–135, Tokyo, Japan. Association for Computational Linguistics.

Yifan Gao, Qingyu Yin, Zheng Li, Rui Meng, Tong Zhao, Bing Yin, Irwin King, and Michael Lyu. 2022. Retrieval-augmented multilingual keyphrase generation with retriever-generator iterative training. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1233–1246, Seattle, United States. Association for Computational Linguistics.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2022. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

Tanishq Gupta, Mohd Zaki, N. M. Anoop Krishnan, and Mausam. 2021. Matscibert: A materials domain language model for text mining and information extraction.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Khaled Hammouda, Diego Matute, and Mohamed S. Kamel. 2005. Corephrase: Keyphrase extraction for document clustering. In *International workshop on machine learning and data mining in pattern recognition*, pages 265–274.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States. Association for Computational Linguistics.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. Document-level entity-based extraction as template generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5257–5269, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shu Huang and Jacqueline M Cole. 2022. Batterybert: A pretrained language model for battery database enhancement. *Journal of Chemical Information and Modeling*.

Anette Hulth. 2003a. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216–223.

Anette Hulth. 2003b. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, page 216–223, USA. Association for Computational Linguistics.

Anette Hulth and Beáta B. Megyesi. 2006. A study on automatically extracted keywords in text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, page 537–544, USA. Association for Computational Linguistics.

Steve Jones and Mark S. Staveley. 1999. Phrasier: A system for interactive document retrieval using keyphrases. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, page 160–167, New York, NY, USA. Association for Computing Machinery.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26, Uppsala, Sweden. Association for Computational Linguistics.

Youngsam Kim, Munhyong Kim, Andrew Cattle, Julia Otmakhova, Suzi Park, and Hyopil Shin. 2013. Applying graph-based keyword extraction to document retrieval. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 864–868, Nagoya, Japan. Asian Federation of Natural Language Processing.

Mikalai Krapivin, Aliaksandr Autaeu, and Maurizio Marchese. 2009. Large dataset for keyphrases extraction. Technical report, University of Trento.

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

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

Rui Liu, Zheng Lin, Peng Fu, and Weiping Wang. 2020. Reinforced keyphrase generation with bert-based sentence scorer. In *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 1–8.

Rui Liu, Zheng Lin, and Weiping Wang. 2021a. Keyphrase prediction with pre-trained language model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3180–3191.

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2021b. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2641–2651, Copenhagen, Denmark. Association for Computational Linguistics.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. BioGPT: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6). Bbac409.

Yichao Luo, Yige Xu, Jiacheng Ye, Xipeng Qiu, and Qi Zhang. 2021. Keyphrase generation with fine-grained evaluation-guided reinforcement learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 497–507, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rui Meng, Tong Wang, Xingdi Yuan, Yingbo Zhou, and Daqing He. 2023. General-to-specific transfer labeling for domain adaptable keyphrase generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1602–1618, Toronto, Canada. Association for Computational Linguistics.

Rui Meng, Xingdi Yuan, Tong Wang, Sanqiang Zhao, Adam Trischler, and Daqing He. 2021. An empirical study on neural keyphrase generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4985–5007, Online. Association for Computational Linguistics.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592, Vancouver, Canada. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 317–326, Berlin, Heidelberg. Springer Berlin Heidelberg.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 40:211–218.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Dhruva Sahrawat, Debanjan Mahata, Mayank Kulkarni, Haimin Zhang, Rakesh Gosangi, Amanda Stent, Agniv Sharma, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2019. Keyphrase extraction from scholarly articles as sequence labeling using contextualized embeddings.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Hoo-Chang Shin, Yang Zhang, Evelina Bakhturina, Raul Puri, Mostofa Patwary, Mohammad Shoeybi, and Raghav Mani. 2020. BioMegatron: Larger biomedical domain language model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4700–4706, Online. Association for Computational Linguistics.

Min Song, Il Yeol Song, Robert B. Allen, and Zoran Obradovic. 2006. Keyphrase extraction-based query expansion in digital libraries. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '06, page 202–209, New York, NY, USA. Association for Computing Machinery.

Mingyang Song, Yi Feng, and Liping Jing. 2022. 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*, pages 5710–5720, Seattle, United States. Association for Computational Linguistics.

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

Mukund Srinath, Shomir Wilson, and C Lee Giles. 2021. Privacy at scale: Introducing the PrivaSeer corpus of web privacy policies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6829–6839, Online. Association for Computational Linguistics.

Prithvishankar Srinivasan and Santosh Mashetty. Sportsbert. `https://huggingface.co/microsoft/SportsBERT`.

Yixuan Su, Deng Cai, Yan Wang, David Vandyke, Simon Baker, Piji Li, and Nigel Collier. 2021. Non-autoregressive text generation with pre-trained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 234–243, Online. Association for Computational Linguistics.

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

Avinash Swaminathan, Haimin Zhang, Debanjan Mahata, Rakesh Gosangi, Rajiv Ratn Shah, and Amanda Stent. 2020. A preliminary exploration of GANs for keyphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8021–8030, Online. Association for Computational Linguistics.

Yixuan Tang, Weilong Huang, Qi Liu, Anthony K. H. Tung, Xiaoli Wang, Jisong Yang, and Beibei Zhang. 2017. Qalink: Enriching text documents with relevant q&a site contents. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*.

Amalie Trewartha, Nicholas Walker, Haoyan Huo, Sanghoon Lee, Kevin Cruse, John Dagdelen, Alexander Dunn, Kristin A. Persson, Gerbrand Ceder, and Anubhav Jain. 2022. Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science. *Patterns*, 3(4):100488.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xiaojun Wan and Jianguo Xiao. 2008a. CollabRank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 969–976, Manchester, UK. Coling 2008 Organizing Committee.

Xiaojun Wan and Jianguo Xiao. 2008b. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, page 855–860. AAAI Press.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Ian Witten, Gordon Paynter, Eibe Frank, Carl Gutwin, and Craig Nevill-Manning. 1999. In *Proceedings of the fourth ACM conference on Digital libraries*, pages 254–255. [link].

Di Wu, Wasi Ahmad, and Kai-Wei Chang. 2023. Rethinking model selection and decoding for keyphrase generation with pre-trained sequence-to-sequence models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore. Association for Computational Linguistics.

Di Wu, Wasi Ahmad, Sunipa Dev, and Kai-Wei Chang. 2022. Representation learning for resource-constrained keyphrase generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 700–716, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Huanqin Wu, Wei Liu, Lei Li, Dan Nie, Tao Chen, Feng Zhang, and Di Wang. 2021. UniKeyphrase: A unified extraction and generation framework for keyphrase prediction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 825–835, Online. Association for Computational Linguistics.

Xiaoyuan Wu and Alvaro Bolivar. 2008. Keyword extraction for contextual advertisement. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, page 1195–1196, New York, NY, USA. Association for Computing Machinery.

Hai Ye and Lu Wang. 2018. Semi-supervised learning for neural keyphrase generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4142–4153, Brussels, Belgium. Association for Computational Linguistics.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2Set: Generating diverse keyphrases as a set. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4598–4608, Online. Association for Computational Linguistics.

Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Peter Brusilovsky, Daqing He, and Adam Trischler. 2020. One size does not fit all: Generating and evaluating variable number of keyphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7961–7975, Online. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9054–9065. Curran Associates, Inc.

Mike Zhang, Kristian Jensen, Sif Sonniks, and Barbara Plank. 2022. SkillSpan: Hard and soft skill extraction from English job postings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4962–4984, Seattle, United States. Association for Computational Linguistics.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on Twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 836–845, Austin, Texas. Association for Computational Linguistics.

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

Guangzhen Zhao, Guoshun Yin, Peng Yang, and Yu Yao. 2022. Keyphrase generation via soft and hard semantic corrections. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7757–7768, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jing Zhao and Yuxiang Zhang. 2019. Incorporating linguistic constraints into keyphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5224–5233, Florence, Italy. Association for Computational Linguistics.