Data Augmentation for Low-Resource Keyphrase Generation

Krishna Garg Jishnu Ray Chowdhury Cornelia Caragea
Computer Science
University of Illinois Chicago
kgarg8@uic.edu jraych2@uic.edu cornelia@uic.edu

Abstract

Keyphrase generation is the task of summarizing the contents of any given article into a few salient phrases (or keyphrases). Existing works for the task mostly rely on large-scale annotated datasets, which are not easy to acquire. Very few works address the problem of keyphrase generation in low-resource settings, but they still rely on a lot of additional unlabeled data for pretraining and on automatic methods for pseudo-annotations. In this paper, we present data augmentation strategies specifically to address keyphrase generation in purely resource-constrained domains. We design techniques that use the full text of the articles to improve both present and absent keyphrase generation. We test our approach comprehensively on three datasets and show that the data augmentation strategies consistently improve the state-of-the-art performance. We release our source code at https://github.com/kgarg8/kpgen-lowres-data-aug.

1 Introduction

Keyphrase generation (KG) helps in document understanding by summarizing the document in the form of a few salient phrases (or keyphrases). These keyphrases may or may not appear verbatim in the original text and accordingly, they are referred to as either *present* or *absent* keyphrases. The task has useful applications to many downstream tasks, e.g., document clustering (Hammouda et al., 2005), matching reviewers to appropriate papers in the conference portals (Augenstein et al., 2017), recommendation systems (Augenstein et al., 2017), text classification (Wilson et al., 2005; Hulth and Megyesi, 2006; Berend, 2011), index construction (Ritchie et al., 2006) and sentiment analysis and opinion mining (Wilson et al., 2005; Berend, 2011).

Prior works for keyphrase generation have largely focused on using large-scale annotated datasets for various domains - computer science (KP20k), news (KPTimes, JPTimes), webpages (OpenKP), etc. However, such large annotation datasets are not available in all domains (e.g., medicine, law, finance), either due to paucity in terms of available data or lack of domain expertise among the annotators or even the high annotation costs. This necessitates the focus on the low-resource domains.

The traditional ways to address low-resource keyphrase generation have been centered around using semi-supervised or unsupervised learning techniques (Ye and Wang, 2018; Wu et al., 2022; Ray Chowdhury et al., 2022). For these methods, a lot of unlabeled data is necessary and needs to be curated for model training. The unlabeled data is further annotated automatically using keyphrase extraction methods and is used for pretraining the model or is used in the auxiliary task for multitasking. There are two limitations to these methods: (1) they have to still depend on additional large-scale unlabeled data, which may not be available always; and (2) the automatic annotation may not be accurate enough, especially when the off-the-shelf keyphrase generation or extraction models are pretrained on a different domain.

In this paper, we develop data augmentation strategies for purely low-resource domains, which do not require acquiring unlabeled data for pretraining or automatic annotation approaches for unlabeled data (which may introduce errors). Inspired by Garg et al. (2022) who showed the benefits of using information beyond the title and abstract for keyphrase generation, we leverage the full text of the documents (which is often ignored by prior works) and present ways for augmenting the text for improving both present and absent keyphrase generation performance.

Data augmentation in NLP has recently become a promising line of research to improve the state-of-the-art performance (Wei and Zou, 2019; Fadaee et al., 2017; Li and Caragea, 2021; Sun et al.,

Findings of the Association for Computational Linguistics: ACL 2023, pages 8442–8455
July 9-14, 2023 ©2023 Association for Computational Linguistics

Methods	Excerpts from different data augmentation methods
Title || Abstract	casesian : a knowledge-based system using statistical and experiential perspectives for improving the knowledge sharing in the medical prescription process [SEP] objectives : knowledge sharing is crucial for better patient care in the healthcare industry
Aug_TA_SR	casesian : a knowledge based system using statistical and experiential perspectives for better the knowledge sharing in the medical examination prescription [SEP] objectives : knowledge sharing is crucial for advantageously patient role care in the healthcare industry
Aug_TA_BT	cassian : a knowledge-based system that uses statistical and experiential perspectives to improve the sharing of knowledge in the medical prescription process [SEP] objectives : knowledge sharing is essential to improve patient care in the health sector
Aug_TA_KPD	casesian : a [MASK] using statistical and experiential perspectives for improving the [MASK] in the [MASK] process [SEP] objectives : [MASK] is crucial for better patient care in the healthcare industry
Aug_TA_KPSR	casesian : a cognition based system using statistical and experiential perspectives for improving the noesis sharing in the checkup prescription process [SEP] objectives : noesis sharing is crucial for better patient care in the healthcare industry
Aug_Body	numerous methods have been investigated for improving the knowledge sharing process in medical prescription [SEP] case-based reasoning is one of the most prevalent knowledge extraction methods
Gold Keyphrases	case-based reasoning , medical prescription , knowledge-based system , knowledge sharing , bayesian theorem

Table 1: An example depicting different augmentation methods used in the paper. The text is highlighted as follows: DIVERSITY introduced in the augmented samples , ABSENT KEYPHRASES , PRESENT KEYPHRASES (highlighted only in Title || Abstract for brevity). Note that all Aug prefixed methods augment as a separate article to the original article T || A. For specific details about each method, please refer to §3.2. Best viewed in color.

2020; Xie et al., 2020; Feng et al., 2020; Park and Caragea, 2022; Yadav and Caragea, 2022). An ideal data augmentation technique is desirous to have the following characteristics: (1) to introduce diversity in training samples but neither too much (otherwise, training samples fail to represent the given domain) nor too less (otherwise, it leads to overfitting); (2) to be easy-to-implement; and (3) to improve model performance.

Towards this end, we design and experiment with four data augmentation techniques (the first two being specifically designed for keyphrase generation) that remake the body[1] of a given article and then augment it to the training data samples containing Title and Abstract (T || A): (1) Aug-Body-KPD where the new training samples contain masked body (i.e., we drop present keyphrases with a certain probability from the body), (2) Aug-Body-KPSR where all the instances of present keyphrases (in contrast to random tokens as in the standard synonym replacement) in the body are replaced with their synonyms, (3) Aug-Body-BT where the body text is translated to an intermediate language and then back to the original language, (4) Aug-Body-SR where the standard synonym replacement is applied to random tokens of the body.

In addition to augmentation with the body, we also provide methods for augmentation using T || A. We depict the representative augmentation strategies in Table 1.

The intuition is that while augmenting the text if we further drop some of the present keyphrases, similar to Masked Language Modeling (Devlin et al., 2019), that makes the task harder and the model is forced to learn to generate the keyphrases. Introducing synonyms and back-translation further increases the diversity of the samples in a much controlled way. Recently, several full-text datasets have been proposed for the KG task, e.g., FullTextKP (Garg et al., 2022), LDKP3K (Mahata et al., 2022), and LDKP10K (Mahata et al., 2022). We use two of these datasets, i.e., LDKP3K and LDKP10K, that contain scientific papers, along with a third dataset KPTimes (Gallina et al., 2019) which mimics full-text keyphrase generation datasets but from a different domain, i.e., news. Through extensive experiments on the three datasets, we observe that although it is hard to improve the present keyphrase generation performance without sacrificing the absent keyphrase generation performance, our proposed augmentation approaches with the body consistently improve both. Moreover, the augmentation methods with body steadily surpass the performance of data augmentation methods that use only Title and Abstract.

[1] Body refers to the full text of the article excluding Title and Abstract.

In summary, the main contribution of the paper is to demonstrate data augmentation strategies for the keyphrase generation task particularly for purely low-resource domains (which have been under-explored). We present simple yet effective data augmentation methods using the full text of the articles and demonstrate large improvements over the state-of-the-art methods.

2 Related Work

Meng et al. (2017) first proposed to solve Keyphrase Generation as a sequence-to-sequence task using deep learning (encoder-decoder) methods. They proposed CopyRNN which uses the copy mechanism (Gu et al., 2016) with the GRU-based encoder-decoder model. This was further extended by Chen et al. (2018) to incorporate correlations between the predicted keyphrases (CorrRNN) and by Yuan et al. (2020) to propose a mechanism to generate a sequence of a variable number of keyphrases (catSeq). Several other works approached the task using reinforcement learning (Chan et al., 2019), generative adversarial networks (Swaminathan et al., 2020), and hierarchical decoding (Chen et al., 2020). Ye et al. (2021b) further reframed the task as sequence-to-set generation instead of sequence-to-sequence generation and used the transformer model for the first time for this task. Later, Garg et al. (2022); Wu et al. (2022); Kulkarni et al. (2022); Wu et al. (2021) used other pre-trained models like Longformer Encoder-Decoder, BART, KeyBART, and UniLM. In this paper, we constrain our focus to *CatSeq* model (Yuan et al., 2020) and explore data augmentation strategies using CatSeq on three datasets. However, our augmentation strategies can be extended to work with other pre-trained models in future work.

Data Augmentation & Keyphrase Generation. Data augmentation has been explored in related tasks like Named-Entity Recognition (Dai and Adel, 2020; Wang and Henao, 2021), and Keyphrase Extraction (Veyseh et al., 2022; Liu et al., 2018), but there have been minimal efforts for exploring data augmentation in Keyphrase Generation. Most of such works deal with augmentation of the candidate keyphrases (extracted using an off-the-shelf unsupervised keyphrase extraction method) to the ground truth keyphrases. Ye and Wang (2018) generated synthetic ground truth labels for the additional unlabeled data. Shen et al. (2022) generated silver labels in addition to the gold-labeled keyphrases using an automatic comparison and ranking mechanism. Chen et al. (2019); Santosh et al. (2021) augmented keyphrases from semantically similar documents to improve keyphrase generation. In contrast, we deal mainly with the augmentation on the input side (i.e., augmenting text to the given articles instead of augmenting the ground-truth keyphrases). Garg et al. (2022) used external information from various parts of the body and appended it to the T || A of the given articles. Our data augmentation strategy is weakly inspired by this work and we use this work as one of the baselines for comparison. Ray Chowdhury et al. (2022) proposed a data augmentation strategy similar to one of our augmentation methods (suffixed with KPD), i.e., randomly dropping *present* keyphrases from text.. We leverage the strategy further to drop the present keyphrases from even the body of the articles and then augment it to the articles themselves.

Low-Resource Keyphrase Generation. Wu et al. (2022) presented a method for a low-resource setting where they utilized the major fraction of a large-scale dataset (KP20k) as unlabeled data for pretraining (using sophisticated pretraining objectives) and the smaller fraction of the dataset for fine-tuning. Ye and Wang (2018) proposed a semi-supervised technique where they created synthetic keyphrases for the large-scale unlabeled data and also utilized the unlabeled data for training the model in a multi-tasking fashion. In contrast, our methods do not require acquiring any unlabeled data or pretraining or multi-task training but work with a few annotated samples. However, all the above works can very well complement our methods to further improve the performance.

3 Methods

In this section, we first describe the formulation of the keyphrase generation task. Next, we describe the baselines followed by the data augmentation strategies that we propose for keyphrase generation.

Problem Formulation. Keyphrase Generation can be posited as a sequence-to-sequence generation task where the input is the text from a given article and the output is a sequence of keyphrases that summarize the article. Formally[2], the task can

[2] We model the problem similar to CATSEQ as proposed by Yuan et al. (2020).

Datasets	#Train	#Dev	#Test	Avg #words	Avg #kp	Avg kp-len	% Present	% Absent
LDKP3K♠	50,000	3,339	3,413	6,457	4.45	1.86	84.24	15.76
LDKP10K♠	50,000	10,000	10,000	4,674	5.98	2.07	74.40	25.60
KPTimes	259,923	10,000	20,000	948	4.03	2.17	48.44	51.56

Table 2: Statistics of the datasets. *#words*: number of words in the document, *#kp*: number of keyphrases, *kp-len*: keyphrase length, % Present (Absent): percentage of present (absent) keyphrases. ♠ indicates *medium* version of the original dataset. Note that the statistics are computed for the combined train, dev and test sets.

be denoted as follows:

Input: Title || Sent$_1$ || Sent$_2$ ||...|| Sent$_k$
Output: kp_1 || kp_2 ||...||kp_n

where kp_i denotes a keyphrase, *Sent$_j$* denotes a sentence from the abstract or from the body of the article, || denotes any delimiter (e.g., [SEP] in this work).

3.1 Baselines

T || A: This baseline contains all the training samples with Title and Abstract concatenated as T [SEP] A.

T || A || Body: For this baseline, we simply concatenate the body of the article to T || A. This baseline was presented in the prior work by Garg et al. (2022).

3.2 Data Augmentation Strategies

Further, as discussed in §1, we describe the data augmentation strategies created primarily using four ways of augmentation: dropout, synonym replacement (both keyphrase-specific and standard) and back-translation. We describe them as follows:

Aug_Body: In this method, we augment the training set with the text from the body of each article, which doubles the total number of samples. That is, one sample is **T** || **A** and the other is **Body** (i.e., sentences from the body of the article).

Aug_Body_KPD: In this method, we first apply the dropout technique presented by Ray Chowdhury et al. (2022) to the body of the article and then augment it (as above). The dropout technique is to mask some of the present keyphrases (particularly, all occurrences of a given keyphrase) in the body of the article.

Aug_TA_KPD: In this method of augmentation, we first apply the dropout technique to the T || A, and then add it to the training set.

Aug_Body_KPSR: In this method, we replace all the present keyphrases in the body of the article with the corresponding synonyms from NLTK WordNet (Miller, 1995) and augment it to the training set. If a particular keyphrase does not have a corresponding synonym, we retain the original keyphrase. Notably, only a small number of keyphrases lack synonyms in the WordNet. For instance, we were able to find synonyms for 2936 (out of 3282) keyphrases for data augmentation on the Body, with 1000 samples of LDKP3K dataset. We show the statistics for the LDKP3K dataset in Table 3.

	1000	2000	4000	8000
Aug_TA_KPSR	3386/	6705/	13374/	26757/
	3733	7385	14702	29398
Aug_Body_KPSR	2936/	5844/	11671/	23538/
	3282	6515	13001	16171

Table 3: Statistics of the synonyms replaced/ total synonyms by Aug_Body_KPSR and Aug_TA_KPSR methods for LDKP3K dataset for four settings, i.e., 1000, 2000, 4000, 8000 samples.

Aug_TA_KPSR: This is similar to Aug_Body_KPSR but with the difference that we replace present keyphrases with their synonyms in the T || A instead of the body of the article.

Aug_Body_BT: In this method, we backtranslate the body of the article from English to French and back to English using Opus-MT (Tiedemann and Thottingal, 2020) pretrained translation models. The backtranslated (or equivalently, paraphrased) articles are then augmented as separate samples to the training set. During the translation of text from one language to another, we use temperature sampling with a temperature value equal to 0.7.

Aug_TA_BT: This method applies back translation model to the T || A instead of the body and does augmentation similar to Aug_Body_BT.

Aug_Body_SR: We use the standard synonym replacement, i.e., we randomly select 10% of the tokens from the body of a given article, replace them with their corresponding synonyms from NLTK Wordnet, and augment the text as a separate article to the training set.

AUG_TA_SR: We do augmentation similar to AUG_BODY_SR but use the T || A instead of body.

4 Experimental Setup

4.1 Datasets

We conduct experiments on three datasets for keyphrase generation. All these datasets contain the full text of the articles along with the keyphrase annotations. 1) **LDKP3K** (Mahata et al., 2022) contains computer science research articles from online digital libraries like ACM Digital Library, ScienceDirect and Wiley. It is a subset of KP20K corpus (Meng et al., 2017) but each article now contains the full text instead of just the title and abstract. 2) **LDKP10K** (Mahata et al., 2022) expands a subset of articles from OAGkx dataset (Çano and Bojar, 2019) to contain their full text. The articles are scientific publications curated from various domains. We use the *medium* version of both LDKP datasets (each consists of 50,000 samples in the training set) to facilitate quality sampling of the articles for the low-resource setting while being mindful of the computational budget. 3) **KPTimes** (Gallina et al., 2019) is a large-scale dataset with long news texts. To mimic KG datasets, we map the heading of the news article to *Title*, and segment the main body of the news article into a maximum of 300-words[3] *Abstract* and the rest of the text as *Body*. We choose KPTimes to validate our observation on an altogether different domain. Datasets' statistics are shown in Table 2. Dataset preprocessing steps are outlined in Appendix §A.

4.2 Evaluation

We compare the performance of the different methods comprehensively for four low-resource settings, i.e., with 1000, 2000, 4000 and 8000 samples. The settings are highly competitive to the prior works where they used at best 5000 samples (Ray Chowdhury et al., 2022; Wu et al., 2022) for their experiments. Following prior works (Meng et al., 2017; Chen et al., 2018; Chan et al., 2019; Chen et al., 2020), we report the results for metrics F1@5[4] and F1@M in the main tables. All comparisons are done after stemming the text as well as keyphrases.

Following Meng et al. (2017); Chan et al. (2019); Yuan et al. (2020), we use GRU encoder-decoder-based architecture for evaluating all models. For all experiments, we restrict the length of the body (or equivalently, full text) to a maximum sequence length of 800 words. For each setting, we sample thrice and further repeat each sample for three different seeds. We thus report the average result for a total of nine runs (3 samples * 3 seeds) for each setting. Hyperparameters and other implementation details are presented in Appendix §A.

5 Results and Analysis

We present our discussion of results for the generation of the two types of keyphrases, i.e., *present* and *absent* in §5.1 and §5.2, respectively.

5.1 Present Keyphrase Generation

From Table 4, we make the following observations. First, augmenting the baseline T || A with the text from the body (AUG_BODY) helps to improve the present keyphrase generation performance. Second, we observe that the methods that use the body (prepended with AUG_BODY) are better than the augmentation methods that just use Title and Abstract (prepended with AUG_TA). These two observations imply that the body constitutes a rich source of present keyphrases.

Third, we also compare with Garg et al. (2022) (T || A || BODY) where they concatenated different types of sentences to T || A. We observe that augmenting the text from the articles (AUG_BODY) instead of merely concatenating them (T || A || BODY) improves the performance by a wide margin. It is also interesting to observe that T || A || BODY, which found significant performance gains in large-scale settings, underperforms even T || A in many purely low-resource settings.

Fourth, the results suggest a quite intriguing observation that the standard data augmentation techniques like synonym replacement and back translation (suffixed with SR, BT) are more rewarding for present keyphrase generation performance than the techniques specifically designed for the keyphrase generation task (suffixed with KPD, KPSR). This trend could be because synonym replacement and back translation bring more diversity to the training samples (since they replace/ rephrase a much larger portion of the text) compared to keyphrase-specific techniques which modify only a handful of tokens (i.e., present keyphrases) in the text. It is worth mentioning that even these standard data augmentation techniques have been largely ignored by the current research on keyphrase generation.

[3] The length was chosen on a similar scale as the average length of abstracts in LDKP10K, which is about 260 words.

[4] We use the metrics from (Chan et al., 2019) and adopted by Chen et al. (2020); Ahmad et al. (2021); Ye et al. (2021a).

LDKP3K	1,000		2,000		4,000		8,000	
	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
T || A	4.68_1	9.10_6	6.19_1	11.89_2	9.67_2	18.47_8	11.97_1	22.86_1
T || A || Body	4.94_1	9.55_5	5.99_2	11.61_2	10.14_1	19.57_0	12.30_0	**23.53_0**
AUG_TA_SR	4.75_1	9.34_3	6.66_2	12.74_0	9.19_3	17.65_{10}	11.37_0	21.95_0
AUG_TA_BT	4.41_1	8.62_2	6.32_2	12.27_3	10.42_0	19.96_1	**12.34_0**	23.32_2
AUG_TA_KPD	4.67_1	9.19_1	6.00_0	11.63_1	7.92_2	15.48_5	10.53_0	20.55_1
AUG_TA_KPSR	4.55_0	8.95_1	5.70_1	10.90_1	7.14_1	13.87_5	9.33_0	18.29_1
AUG_Body	**5.33_2**	**10.42_5**	**7.10_6**	**13.92_{18}**	9.97_5	19.25_{18}	11.82_2	22.67_4
AUG_Body_SR	4.88_1	9.69_4	6.50_0	12.53_2	9.36_9	18.15_{30}	12.19_1	23.04_3
AUG_Body_BT	4.59_0	9.04_2	6.36_3	12.26_5	**10.50_0**	**20.09_1**	12.31_1	23.19_3
AUG_Body_KPD	4.72_2	9.31_6	6.12_1	11.92_3	8.82_7	17.04_{18}	11.61_0	22.14_1
AUG_Body_KPSR	4.60_0	9.15_1	5.78_1	11.21_6	7.44_2	14.60_8	11.40_1	21.64_3

LDKP10K	1,000		2,000		4,000		8,000	
	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
T || A	**4.47_1**	8.27_3	6.66_1	12.32_2	9.95_1	17.49_2	11.31_1	19.76_3
T || A || Body	3.89_4	7.30_{14}	6.55_1	12.07_1	9.81_0	17.54_1	11.70_1	20.24_2
AUG_TA_SR	4.37_0	8.26_0	6.22_0	11.67_1	**10.69_0**	**18.33_0**	**12.30_0**	**20.86_0**
AUG_TA_BT	4.04_1	7.59_2	**7.70_1**	**14.01_3**	10.27_1	18.00_2	10.44_0	18.26_0
AUG_TA_KPD	3.79_0	7.18_2	5.14_1	9.75_3	9.53_3	16.68_6	11.29_2	19.92_5
AUG_TA_KPSR	3.74_0	7.11_1	4.77_1	9.08_3	8.66_3	15.19_6	10.22_0	17.78_1
AUG_Body	4.45_6	**8.45_{21}**	6.98_5	12.90_{12}	10.37_0	18.28_0	11.92_1	20.73_3
AUG_Body_SR	4.22_0	8.01_0	6.36_0	11.88_1	10.12_0	17.91_0	11.57_0	20.38_1
AUG_Body_BT	4.38_0	8.17_2	7.65_3	13.87_5	10.24_0	17.95_1	11.17_1	19.73_3
AUG_Body_KPD	3.96_3	7.49_8	5.61_2	10.54_4	9.43_0	16.81_0	11.03_0	19.53_0
AUG_Body_KPSR	3.83_0	7.24_1	4.77_1	9.06_2	9.24_0	16.49_0	10.93_0	19.27_0

KPTimes	1,000		2,000		4,000		8,000	
	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
T || A	9.83_0	19.01_1	13.49_3	24.49_7	16.92_1	28.84_0	19.13_0	31.89_0
T || A || Body	9.66_2	18.56_6	13.64_2	24.98_4	16.74_0	29.33_0	18.91_0	32.20_0
AUG_TA_SR	11.20_{10}	21.17_{19}	**15.21_0**	26.59_0	**17.30_0**	29.36_0	19.30_0	32.44_0
AUG_TA_BT	11.02_4	21.22_8	13.93_3	25.99_7	16.31_2	29.29_3	18.71_1	32.69_0
AUG_TA_KPD	8.94_0	17.41_0	12.90_1	23.63_3	15.58_1	27.86_0	17.64_1	30.91_1
AUG_TA_KPSR	9.12_2	17.88_4	13.83_2	24.90_2	15.77_0	27.60_0	17.99_0	30.93_0
AUG_Body	9.78_3	19.62_{11}	14.31_1	26.25_1	17.26_1	**30.33_1**	**19.39_1**	33.01_1
AUG_Body_SR	**11.21_4**	**22.05_7**	14.46_1	**26.78_1**	16.86_1	30.13_1	18.96_0	**33.23_0**
AUG_Body_BT	10.46_2	20.24_0	14.12_0	25.92_0	16.46_0	29.28_1	18.88_3	32.75_1
AUG_Body_KPD	8.80_3	17.62_9	13.36_2	24.76_3	16.49_1	29.43_1	18.48_1	32.17_0
AUG_Body_KPSR	10.21_3	20.27_8	13.62_0	25.82_0	16.25_1	29.51_2	18.02_1	32.34_1

Table 4: Performance for generation of present keyphrases. The results are highlighted with blue (↑) and red (↓) with respect to baseline T || A. || denotes concatenation of the text. Standard deviation is subscripted to each number and is reported as a multiple of ± 0.001. Best viewed in color.

Fifth, we rather observe that the keyphrase-specific data augmentation techniques are not just lower in performance than the standard data augmentation techniques but often they hurt the performance of the model when trained in purely low-resource settings. The reason could be that the models do not have enough samples and diversity to learn to generate the present keyphrases, all the more when the present keyphrases are dropped or replaced during training. This is in contrast with the behavior of models when trained on a large-scale dataset, where the performance of present keyphrase generation (AUG_TA_KPD) is on par with T || A (Ray Chowdhury et al., 2022).

Sixth, in Table 4, we can also compare the performance of models trained on: (1) total x original samples, (2) x original + x augmented samples, (3) total $2x$ original samples. For example, for LDKP3K dataset, we observe that 2000 original samples achieve the best performance (11.89 in F1@M), followed by the augmented version (9.34 for augmentation with synonym replacement, 10.42 for augmentation with body) whereas the performance when using 1000 original samples is 9.10. We observe similar trends across the different augmentation strategies and datasets.

We draw the following conclusions: (1) Data augmentation techniques for keyphrase generation have been quite an under-studied topic, particularly for low-resource settings and the behavior of the models is different than that when training on large-scale settings; (2) We show that existing works such as those by Garg et al. (2022); Ray Chowdhury et al. (2022) can be surpassed by the data augmentation methods discussed in this work when used in low-resource settings for *present* keyphrase generation.

LDKP3K	1,000		2,000		4,000		8,000	
	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
T || A	0.078_0	0.169_0	0.129_0	0.281_0	0.044_0	0.093_0	0.044_0	0.099_0
T || A || Body	0.079_0	0.165_0	0.130_0	0.282_0	0.047_0	0.105_0	0.031_0	0.073_0
AUG_TA_SR	0.132_0	0.290_0	0.136_0	0.300_0	0.096_0	0.207_0	0.067_0	0.141_0
AUG_TA_BT	0.128_0	0.279_0	0.139_0	0.305_0	0.068_0	0.140_0	0.121_0	0.266_0
AUG_TA_KPD	0.140_0	0.311_0	0.145_0	0.318_0	0.141_0	0.307_0	0.099_0	0.218_0
AUG_TA_KPSR	0.142_0	0.307_0	0.177_0	0.393_0	0.151_0	0.321_0	0.154_0	0.325_0
AUG_Body	0.129_0	0.291_0	0.130_0	0.292_0	0.061_0	0.138_0	0.079_0	0.175_0
AUG_Body_SR	0.141_0	0.319_0	0.157_0	0.342_0	0.076_0	0.161_0	0.149_0	0.322_0
AUG_Body_BT	0.130_0	0.287_0	0.121_0	0.265_0	0.081_0	0.183_0	0.120_0	0.253_0
AUG_Body_KPD	0.144_0	0.328_0	0.189_0	0.407_0	0.136_0	0.298_0	0.182_0	0.398_0
AUG_Body_KPSR	**0.162_0**	**0.359_0**	**0.200_0**	**0.441_0**	**0.184_0**	**0.405_0**	**0.227_0**	**0.495_0**

LDKP10K	1,000		2,000		4,000		8,000	
	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
T || A	0.023_0	0.047_0	0.039_0	0.079_0	0.114_0	0.228_0	0.184_0	0.335_0
T || A || Body	0.021_0	0.044_0	0.035_0	0.074_0	0.052_0	0.105_0	0.159_0	0.289_0
AUG_TA_SR	0.031_0	0.061_0	0.054_0	0.110_0	0.195_0	0.387_0	0.355_0	0.629_0
AUG_TA_BT	0.027_0	0.051_0	0.084_0	0.173_0	0.196_0	0.383_0	0.337_0	0.617_0
AUG_TA_KPD	0.020_0	0.041_0	0.057_0	0.115_0	0.210_0	0.403_0	0.299_0	0.552_0
AUG_TA_KPSR	0.031_0	0.059_0	0.067_0	0.133_0	0.229_0	0.433_0	0.429_0	0.769_0
AUG_Body	0.033_0	0.063_0	0.071_0	0.148_0	0.206_0	0.407_0	0.344_0	0.622_0
AUG_Body_SR	0.037_0	0.071_0	0.085_0	0.168_0	0.213_0	0.410_0	0.378_0	0.687_0
AUG_Body_BT	0.033_0	0.064_0	0.073_0	0.151_0	0.193_0	0.387_0	0.338_0	0.637_0
AUG_Body_KPD	0.044_0	0.088_0	0.085_0	0.166_0	0.238_0	0.465_0	0.400_0	0.726_0
AUG_Body_KPSR	**0.045_0**	**0.089_0**	**0.106_0**	**0.210_0**	**0.259_0**	**0.492_0**	**0.459_0**	**0.827_0**

KPTimes	1,000		2,000		4,000		8,000	
	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M	F1@5	F1@M
T || A	0.026_0	0.051_0	0.026_0	0.247_0	1.430_0	2.445_1	3.066_0	5.393_1
T || A || Body	0.023_0	0.044_0	0.023_0	0.271_0	1.082_0	1.950_0	2.558_0	4.719_0
AUG_TA_SR	0.105_0	0.176_0	1.240_0	2.168_0	2.718_0	4.648_0	4.274_0	7.336_0
AUG_TA_BT	0.163_0	0.277_0	0.163_1	2.050_2	2.501_0	4.390_0	3.826_1	6.818_1
AUG_TA_KPD	0.060_0	0.107_0	0.060_0	0.637_0	1.634_0	2.854_0	3.423_0	6.006_0
AUG_TA_KPSR	0.087_0	0.163_0	0.087_0	1.841_1	**2.748_0**	4.648_0	**4.465_0**	7.352_0
AUG_Body	0.033_0	0.060_0	0.033_0	1.150_0	2.460_0	4.380_0	4.171_0	7.385_0
AUG_Body_SR	0.159_0	0.278_0	1.122_0	1.999_0	2.681_0	4.708_0	4.135_1	7.285_1
AUG_Body_BT	0.131_0	0.239_0	**1.191_0**	**2.152_0**	2.363_0	4.215_1	3.795_0	6.670_1
AUG_Body_KPD	0.038_0	0.069_0	0.038_0	1.200_1	2.588_0	4.607_0	4.382_0	7.575_0
AUG_Body_KPSR	**0.182_0**	**0.319_0**	0.182_0	1.963_1	2.708_0	**4.744_0**	4.325_1	**7.629_2**

Table 5: Performance for generation of absent keyphrases. The results are highlighted with blue (↑) and red (↓) with respect to baseline T || A. || denotes concatenation of the text. Standard deviation is subscripted to each number and is reported as a multiple of ± 0.001. Best viewed in color.

5.2 Absent Keyphrase Generation

To investigate the ability of the KG models to develop a semantic understanding of the documents, we evaluate the performance of the absent keyphrase generation. Table 5 presents the absent keyphrase performance of the different augmentation methods. Our observations are as follows. First, augmentation with the body (prefixed with AUG_BODY) still surpasses the Title and Abstract (prefixed with AUG_TA) counterparts. Second, unlike the present keyphrase generation performance, the absent keyphrase generation performance is generally better with almost all the data augmentation methods compared to the baseline T || A. The reason could be that the augmentation methods artificially turn some of the present keyphrases to absent keyphrases (e.g., present keyphrases replaced with synonyms or dropped or rephrased). Thus, the model finds much more opportunities to learn to generate absent keyphrases.

Third, interestingly, KG-targeted data augmentation methods (suffixed with KPD, KPSR) perform better than the standard data augmentation methods like synonym replacement and back translation (suffixed with SR, BT) for generating absent keyphrases (unlike present keyphrase generation). This is because KPD, KPSR specifically replace the present keyphrases to become absent keyphrases. Whereas SR, BT *randomly* replace/ rephrase the tokens and thus, one would expect a less number of present keyphrases turning into absent keyphrases. Fourth, augmentation with KG-based synonym replacement (KPSR) surpasses even the dropout augmentation technique (KPD). This might be because of two reasons: (1) the keyphrase dropout method masks the keyphrases

Excerpts from test dataset samples	Methods	Predicted Keyphrases
committees of learning agents [SEP] we describe how machine learning and decision theory is combined in an application that supports control room operators of a combined heating and power plant ... **Gold:** machine learning ; committees ; decision analysis	T || A	learning
	Aug_Body	machine learning
	Aug_Body_SR	learning
compositional analysis for linear control systems [SEP] the complexity of physical and engineering systems , both in terms of the governing physical phenomena and the number of subprocesses involved ... **Gold:** compositional reasoning ; linear systems ; simulation relations ; assume-guarantee reasoning	T || A	control
	Aug_Body	linear control; linear systems
	Aug_Body_SR	linear control; linear systems
the bits and flops of the n-hop multilateration primitive for node localization problems [SEP] the recent advances in mems , embedded systems and wireless communication technologies are making the realization ... **Gold:** technologies ; ad-hoc localization ; sensor networks ; embedded systems ; wireless ; network	T || A	tangible
	Aug_Body	wireless networks
	Aug_Body_SR	sensors

Table 6: Sample predictions using models trained with different (representative) augmentation methods and the baseline (T || A). The text is highlighted as follows: Present Keyphrases , Absent Keyphrases . Note that the test samples contain only T || A. Best viewed in color.

with some probability value whereas we replace all the present keyphrases with their synonyms, (2) dropping the important keyphrases hides some information from the model, while replacing the keyphrases with their synonyms still largely preserves the semantics and integrity of the text.

Fifth, we observe that the model proposed by Garg et al. (2022) which is based on concatenation is not able to generalize well in the low-resource settings, rather, ends up weakening the model performance compared to T || A. This again urges towards the development of data augmentation methods in purely low-data regimes.

Sixth, in Table 5, the results show that the model trained on the combination of original and augmented samples outperforms the settings where the model is trained on equivalent amount of original samples, for most datasets and augmentation strategies. For instance, for LDKP3K dataset, the 2000 augmentation version achieves 0.290 in F1@M (for augmentation with synonym replacement on Title and Abstract) and outperforms both 2000 original samples (0.281) and 1000 original samples (0.169). Thus, for the same amount of data (2000 dataset size), the augmented version shows better results than without data augmentation.

We show sample predictions from the representative models: T || A (baseline), Aug_Body (best for Present KG), Aug_Body_SR (best for Absent KG) in Table 6. In the table, we can observe that while T || A fails to capture the specific topics (or keyphrases) for the document, models trained with augmentation strategies can generalize better.

Methods	Pres.KP	Abs.KP	TotalKP
T || A	3374	2093	5467
T || A || Body	3985	1482	5467
AUG_TA_SR	5761	5173	10934
AUG_TA_BT	5499	5435	10934
AUG_TA_KPD	4586	6348	10934
AUG_TA_KPSR	4532	6402	10934
AUG_Body	**6309**	4625	10934
AUG_Body_SR	5402	5532	10934
AUG_Body_BT	5291	5643	10934
AUG_Body_KPD	4590	**6344**	10934
AUG_Body_KPSR	4591	6343	10934

Table 7: Number of present, absent, total keyphrases in the training set of LDKP3K with 1000 samples for the different augmentation methods.

6 Analysis

In this section, we study one of the settings in more detail, i.e., with the LDKP3K dataset having 1000 samples in the training set (and twice the number in the training set for Aug-prefixed methods). The study unfolds into two aspects: (a) analyzing the data created for the different augmentation methods, (b) developing better inference strategies.

We analyze the data created using the different augmentation methods and report the present, absent and total number of keyphrases in Table 7. First, we observe that all the data augmentation methods have double the total number of keyphrases because the total number of samples is doubled. In effect, the model develops a better generalization ability when it practices with more instances of present and absent keyphrases. Second, we see that Aug_Body has the highest number of present keyphrases. This implies that the text

Methods	Present		Absent	
	F1@5	F1@M	F1@5	F1@M
T || A	4.68	9.10	0.078	0.169
AUG_TA_BT	4.41	8.62	0.128	0.279
AUG_TA_KPSR	4.55	8.95	0.132	0.290
AUG_Body	**5.33**	**10.42**	0.129	0.291
AUG_Body_BT	4.59	9.04	0.130	0.287
AUG_Body_KPD	4.72	9.31	0.144	0.328
AUG_Body_KPSR	4.60	9.15	**0.162**	**0.359**
Inference Strategies				
Body ∪ Body-KPSR	6.41	**11.95**	0.196	0.428
TA-BT ∪ Body-BT	5.39	10.19	0.160	0.342
TA-KPSR ∪ Body-KPSR	6.17	11.47	**0.220**	**0.462**
Body-BT ∪ Body-KPD	**6.45**	11.81	0.204	0.435
Body-KPSR ∪ Body-KPD	5.94	11.18	0.204	0.444

Table 8: A comparison of various Inference Strategies using *Union* (see §6) with the individual (AUG_) methods on LDKP3K with 1000 samples in the training set.

from the body of the articles not only adds diversity to the training samples (as also evident from Tables 1, 4), but also the diversity contains a lot of present keyphrases, unlike other augmentation methods like KPD, KPSR. Third, it is also evident from Table 7 that the KG-specific data augmentation methods (suffixed with KPD, KPSR) are rich sources of absent keyphrases whereas the standard data augmentation (suffixed with SR, BT) methods are rich in present keyphrases. This further explains the observations made in the previous sections §5.1-5.2 that the KG-specific augmentation methods perform better for absent keyphrase generation, whereas the standard data augmentation methods do better in present keyphrase generation.

Further, in Table 8, we present some of the representative inference strategies by unionizing different augmentation methods during inference. *Union* can be seen as a post-training augmentation method that (during inference) takes a union of the predictions from multiple models that are pretrained using different augmentation methods. The idea is to leverage the complementary strength of the different models that are good for either or both present and absent keyphrase generation. As expected, the performance of the *Union* methods surpasses that of the individual augmentation methods.

7 Conclusion

Although data augmentation has been a very common practice to advance the state-of-the-art in NLP, it has been under-explored for the keyphrase generation (KG) task. Thus, this work discusses various data augmentation methods including both types (i.e., standard and KG-specific) particularly for purely low-resource keyphrase generation, and provides comprehensive evaluation for 12 different settings (four settings for three datasets each).

We also leverage the full text of the articles for data augmentation and observe large improvements over the baseline as well as over data augmentation methods that use only title and abstract (T || A). Detailed analysis helps us believe that KG-specific data augmentation methods can largely improve absent keyphrase generation but at the cost of present keyphrase generation. In contrast, the standard data augmentation techniques like synonym replacement and back-translation are capable of introducing enough diversity to improve the present keyphrase generation without bringing a drop in absent keyphrase generation performance. Although augmentation with the body improves both types of generation to some degree, this work leaves much room to develop better data augmentation strategies to train the model to do better on both present and absent keyphrase generation in low-resource settings which are prevalent in many domains.

8 Limitations

We conducted extensive experiments with three datasets from different domains to substantiate the results thoroughly. We observe the best performance when we also leverage the body of the articles. So, we did not evaluate the performance on the datasets that do not have the full text (or equivalently, long text) of the articles.

Ethics Statement

The datasets we used in experiments are publicly available. In our work, we provide a comprehensive analysis and present data augmentation strategies specifically to address keyphrase generation in purely resource-constrained domains. We do not expect any direct ethical concern from our work.

Acknowledgments

This research is supported in part by NSF CAREER award #1802358, NSF CRI award #1823292, NSF IIS award #2107518, and UIC Discovery Partners Institute (DPI) award. Any opinions, findings, and conclusions expressed here are those of the authors and do not necessarily reflect the views of NSF or DPI. We thank AWS for computational resources used for this study. We also thank our anonymous reviewers for their valuable and constructive feedback and suggestions.

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.

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 *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. 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.

Erion Çano and Ondřej Bojar. 2019. Keyphrase generation: A multi-aspect survey. In *2019 25th Conference of Open Innovations Association (FRUCT)*, pages 85–94. IEEE.

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. 2019. 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.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Steven Y. Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. 2020. GenAug: Data augmentation for finetuning text generators. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 29–42, Online. 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.

Krishna Garg, Jishnu Ray Chowdhury, and Cornelia Caragea. 2022. Keyphrase generation beyond the boundaries of title and abstract. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5809–5821, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Khaled M Hammouda, Diego N 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. Springer.

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 44th Annual Meeting of the Association for Computational Linguistics*, pages 537–544, Sydney, Australia. Association for Computational Linguistics.

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.

Yingjie Li and Cornelia Caragea. 2021. Target-aware data augmentation for stance detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1850–1860, Online. Association for Computational Linguistics.

Qianying Liu, Daisuke Kawahara, and Sujian Li. 2018. Scientific keyphrase extraction: extracting candidates with semi-supervised data augmentation. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data: 17th China National Conference, CCL 2018, and 6th International Symposium, NLP-NABD 2018, Changsha, China, October 19–21, 2018, Proceedings 17*, pages 183–194. Springer.

Debanjan Mahata, Naveen Agarwal, Dibya Gautam, Amardeep Kumar, Swapnil Parekh, Yaman Kumar Singla, Anish Acharya, and Rajiv Ratn Shah. 2022. Ldkp - a dataset for identifying keyphrases from long scientific documents. *DL4SR-22: Workshop on Deep Learning for Search and Recommendation, co-located with the 31st ACM International Conference on Information and Knowledge Management (CIKM)*.

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.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Seo Yeon Park and Cornelia Caragea. 2022. A data cartography based MixUp for pre-trained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4244–4250, Seattle, United States. Association for Computational Linguistics.

Jishnu Ray Chowdhury, Seo Yeon Park, Tuhin Kundu, and Cornelia Caragea. 2022. KPDROP: Improving absent keyphrase generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4853–4870, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Anna Ritchie, Simone Teufel, and Stephen Robertson. 2006. How to find better index terms through citations. In *Proceedings of the Workshop on How Can Computational Linguistics Improve Information Retrieval?*, pages 25–32, Sydney, Australia. Association for Computational Linguistics.

TYSS Santosh, Debarshi Kumar Sanyal, Plaban Kumar Bhowmick, and Partha Pratim Das. 2021. Gazetteer-guided keyphrase generation from research papers. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 655–667. Springer.

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.

Xianjie Shen, Yinghan Wang, Rui Meng, and Jingbo Shang. 2022. Unsupervised deep keyphrase generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11303–11311.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip Yu, and Lifang He. 2020. Mixup-transformer: Dynamic data augmentation for NLP tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3436–3440, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Amir Pouran Ben Veyseh, Nicole Meister, Franck Dernoncourt, and Thien Huu Nguyen. 2022. Improving keyphrase extraction with data augmentation and information filtering. *Association for the Advancement of Artificial Intelligence Workshop*.

Rui Wang and Ricardo Henao. 2021. Unsupervised paraphrasing consistency training for low resource named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5308, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

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.

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.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268.

Shweta Yadav and Cornelia Caragea. 2022. Towards summarizing healthcare questions in low-resource setting. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2892–2905, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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, Ruijian Cai, Tao Gui, and Qi Zhang. 2021a. Heterogeneous graph neural networks for keyphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2705–2715, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021b. 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.

A More Implementation Details

Following Garg et al. (2022), we preprocessed the full text of the articles for all three datasets. We filtered all the articles that had either of the four fields missing, viz., title, abstract, keyphrases, full text, or that contained less than five sentences in the full text. We segmented the full text into sentences using PunktSentenceTokenizer[5] and tokenized the sentences further into tokens using NLTK's word_tokenizer. We also lowercased the text, removed html text, emails, urls, escape symbols, and converted all the numbers into <digit> (Meng et al., 2017), and finally removed any duplicate items in the collection. Further, we sub-sampled the datasets to construct four low-resource settings (sampled thrice for each setting) containing 1000, 2000, 4000 and 8000 samples.

We use the GRU-based architecture for evaluating all the methods. Similar to Meng et al. (2017); Yuan et al. (2020); Chan et al. (2019) we use an encoder-decoder architecture (where both the encoder and the decoder are GRUs) with attention and a pointer mechanism (See et al., 2017). The exact details of the architecture are similar to that of Chan et al. (2019). The vocabulary size is 50,000 and each word is translated into embeddings of dimension equal to 100. The GRU encoders and decoders have hidden layer sizes of 150 and 300 respectively. We use a learning rate of 1e-3, batch size of 4, Adam optimizer, ReduceLROnPlateau scheduler and maximum epochs as 20. We early stop the training with patience value of 2.

[5] https://www.nltk.org/_modules/nltk/tokenize/punkt.html

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
8

☒ A2. Did you discuss any potential risks of your work?
No potential risks

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Left blank.

☑ B1. Did you cite the creators of artifacts you used?
4.1

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
All datasets are open-sourced and we checked the license before using.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
All datasets are open-sourced and we checked the license before using.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4.1

C ☑ Did you run computational experiments?

5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Limitations

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4.1, Appendix, Limitations

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix, 4.2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.