Cross-Domain Robustness of Transformer-based Keyphrase Generation*

Anna Glazkova[1,3][0000−0001−8409−6457] and Dmitry Morozov[2,3][0000−0003−4464−1355]

[1] University of Tyumen, Tyumen, Russia
a.v.glazkova@utmn.ru
[2] Novosibirsk State University, Novosibirsk, Russia
morozowdm@gmail.com
[3] The Institute for Information Transmission Problems (Kharkevich Institute), Moscow, Russia

Abstract. Modern models for text generation show state-of-the-art results in many natural language processing tasks. In this work, we explore the effectiveness of abstractive text summarization models for keyphrase selection. A list of keyphrases is an important element of a text in databases and repositories of electronic documents. In our experiments, abstractive text summarization models fine-tuned for keyphrase generation show quite high results for a target text corpus. However, in most cases, the zero-shot performance on other corpora and domains is significantly lower. We investigate cross-domain limitations of abstractive text summarization models for keyphrase generation. We present an evaluation of the fine-tuned BART models for the keyphrase selection task across six benchmark corpora for keyphrase extraction including scientific texts from two domains and news texts. We explore the role of transfer learning between different domains to improve the BART model performance on small text corpora. Our experiments show that preliminary fine-tuning on out-of-domain corpora can be effective under conditions of a limited number of samples.

Keywords: Keyphrase extraction · BART · Transfer learning · Scholarly document · Text summarization.

1 Introduction

The task of keyphrase generation aims at predicting a set of keyphrases summarizing the content of the source text. Keyphrases are often indexed in databases to improve the performance of information retrieval tools. Researchers select keyphrases for their papers to increase their visibility in the scientific community. Automatic selection of keyphrases for scholarly documents helps to analyze the current research trends, recommend papers, and identify potential peer reviewers [39].

* Supported by the grant of the President of the Russian Federation no. MK-3118.2022.4.

Abstract. The study considers robust estimation of linear regression parameters by the regularization method, the pseudoinverse method, and the Bayesian method allowing for correlations and errors in the data. Regularizing algorithms are constructed and their relationship with pseudoinversion, the Bayesian approach, and BLUE is investigated

Keyphrases: linear regression problems regularization, robust estimation, linear regression parameters, pseudoinverse method, Bayesian method, pseudoinversion, Bayesian approach, BLUE, Bayes methods, estimation theory, probability, statistical analysis

Fig. 1. An example of a source text with the corresponding list of from the Inspec corpus [22]. The keyphrases that appear in the text are underlined.

Figure 1 demonstrates an example of a source text and its keyphrases. Some keyphrases are present in the source text while others are absent. Most unsupervised approaches for keyphrase selection have the purpose of keyphrase extraction, in other words, the ranking and selection of phrases that appear in the text. Recent generative approaches produce both keyphrases present in the text and those absent from it. These approaches utilize deep learning methods using the encoder-decoder architecture [8, 32, 48] and various training techniques, such as incorporating a copying mechanism [44], reinforcement learning [10], hierarchical decoding [11], and multitask learning [26].

Currently, the models for automatic text generation achieve high results in various natural language processing tasks. Since a list of keyphrases is some type of summary of a scientific text, pre-trained models for abstractive summarization appear to be effective for generating keyphrases as a sequence. In our previous work [18, 19], we explored the performance of some of these models for keyphrase generation. It was shown that BART [27] fine-tuned for generating lists of keyphrases on texts from the target domain showed competitive results as compared to several baselines. However, it can show lower performance on the texts from other corpora and domains similar to other fine-tuned models. Our goal is to evaluate whether we can transfer knowledge from the BART model that was fine-tuned to generate keyphrases for one domain to another ones. We seek to answer the following research questions:

- **RQ1.** How effective is a text summarization model fine-tuned on one corpus or one domain for generating keyphrases from texts of other corpora or domains in zero-shot settings?
- **RQ2.** Can we improve the model performance by adding training examples from other corpora and domains?
- **RQ3.** With a small number of training examples, can the model perform as effectively as a model fine-tuned on larger corpora?
- **RQ4.** Can transfer learning improve the model performance using a varying size of training data?

The paper is organized as follows. Section 2 presents related works in the field. In Section 3, we describe the corpora. Section 4 contains a brief description of the models we used. Section 5 presents the experimental setup. The results are reported and discussed in Section 6. Section 7 concludes this paper.

2 Related Work

2.1 Abstractive Text Summarization using Pre-trained Transformers

Pre-trained language models show impressive results in many natural language processing (NLP) tasks. A pre-trained model is a saved network that is previously trained on a large dataset. This is a common and highly effective approach to deep learning on small datasets [15]. Automatic text summarization is a relevant trend in NLP. A summary can be generated through extractive, as well as abstractive, methods. Abstractive methods are difficult to implement because they need a lot of natural language processing. However, abstractive models, such as BART [27], PEGASUS [49], and many others, allow us to generate novel samples by either rephrasing or using new words, instead of simply extracting the important sentences [21, 41].

Neural abstractive summarization based on pre-trained language models has been studied by many researchers and showed a high performance with the aid of large text corpora. In particular, abstractive summarization models were applied for generating summaries in news [5, 13, 20, 52], scientific [6, 35, 45], sport [31], and financial domains [42, 51]. One of the main challenges related to neural abstractive summarization is that domain-shifting problems and overfitting could occur with a small number of samples for the target corpora [12]. The use of additional texts for other corpora is not always successful since different corpora contain texts of different writing styles and forms. The annotation for abstractive summarization is costly. Therefore, exploring approaches to low-resource abstractive summarization is very relevant and attracts the attention of scientists.

2.2 Keyword Selection

Keyword selection approaches can be roughly divided into three categories: i) actual keyword extraction, ii) keyword assignment, and iii) keyphrase generation [1, 8]. The actual keyword extraction involves extracting words directly presented in the text. In keyword assignment, keywords are chosen from a predefined set of terms, while documents are classified into thematic categories according to their topics. Keyphrase generation aims to produce a set or string of keywords using recent advances in sequence-to-sequence applications of neural networks. In this work we focus on keyphrase generation but use some keyword extraction approaches as baselines. Keyphrase generation allows us to generate broad terms and keyphrases that are not presented in the source text in an explicit form.

To date, some scholars have examined neural models to generate multiple keyphrases as a sequence [8, 39]. Chowdhury et al. [14] demonstrated that fine-tuned BART shows competitive results in keyphrase generation compared with the existing extractive neural models. In [23, 46], the authors experimented with controllable text generation for producing keyphrases. The authors of [26] proposed KeyBART, a new pre-training setup for the BART model that learns to generate keyphrases in their original order in the source document. Shen and Le [37] investigated the advantages of title attention and sequence code representing phrase order in a keyphrase sequence in improving Transformer-based keyphrase generation.

The authors of [38] provided a comprehensive survey on recent advances in keyphrase selection from pre-trained language models. They emphasize that most existing keyphrase extraction datasets and studies are based on a few of the most common topics and lack datasets and research related to other domains. Therefore, transferring knowledge from one domain to another to build domain-specific keyphrase extraction models is one of the major challenges for keyphrase generation.

3 Data

The experiments are carried out on six corpora for keyphrase selection:

- Krapivin [25] and Inspec [22] containing scientific texts from the computer science domain;
- PubMed [36] and NamedKeys [17], which include scientific texts from the biomedical domain;
- DUC-2001 [43] and KPTimes [16] consisting of news texts.

The Krapivin corpus contains full papers divided into titles, abstracts, and bodies. In this work, we separately utilized the abstract and body of the paper to select keyphrases (Krapivin-A and Krapivin-T respectively). The original KPTimes corpus is composed of 279,923 article-keyphrase pairs. Here, we used only a test set of the original corpus containing 20,000 samples. Summary statistics for the corpora are presented in Table 1. The most popular keyphrases are shown in Table 2. A comparison of the contents of the corpora is given in Table 3.

4 Models

For keyphrase generation, we utilized BART-base [27], a transformer-based denoising autoencoder for pre-training a seq2seq model. The model has 12 layers, 768 hidden units per layer, and a total of 139M parameters. BART was pre-trained by corrupting documents and then optimizing a reconstruction loss—the cross-entropy between the decoder's output and the original document. We fine-tuned BART-base for six epochs with a maximum sequence length of 256 tokens. We utilized a standard cross-entropy loss and the AdamW optimizer [30]. We

Table 1. A summary statistics for the corpora. The average number of tokens was obtained using NLTK [2]. The "±" sign is utilized to indicate a standard deviation. The abbreviations in this table are CS — computer science, BM — biomedical, A — abstract, and T — text (body).

Characteristic	Krapivin-A	Krapivin-T	Inspec	PubMed	NamedKeys	DUC-2001	KPTimes
Size	2,294	2,293	2,000	1,320	3,049	308	20,000
Domain	scientific, CS			scientific, BM		news	
Type of texts	A	T	A	T	A		
Avg. number of tokens	169.06 ±68.58	8597.63 ±2411.77	127.35 ±65.03	5270.97 ±2690.67	274.67 ±99.88	848.22 ±563.41	733.78 ±477.49
Avg. number of sentences	6.64 ±2.69	343.95 ±107.3	5.3 ±2.73	206.81 ±127.11	10.52 ±3.66	34.74 ±23.33	26.65 ±22.19
Avg. keyphrases per text	5.34±2.77		14.11 ±6.41	5.4 ±2.17	14.15 ±5.2	8.08 ±1.87	5.03 ±1.88
Absent keyphrases, %	51.3 ±25.99	18.04 ±19.69	43.8 ±17.83	13.52 ±19.7	1.04 ±5.83	2.45 ±7.94	35.72 ±29.22
Number of unique keyphrases	8,703		19,066	5,580	20,804	1,850	21,126

used the source text as an input of the model and a list of keyphrases in a string format as an output. Keyphrases included in lists of keyphrases were separated with commas.

As baselines, we used the implementations of TopicRank [4] and YAKE! [7] from the PKE library [3] and KeyBART [26] that represents pre-trained BART-based architecture to produce a sequence of keyphrases pre-trained on the OAGKX dataset [9], which consists of 23 million scientific documents across multiple domains.

5 Experimental Setup

We randomly split each corpus into a 70% training set and a 30% test set. For BART, we performed three runs for each model and then calculated the average results. Since TopicRank and YAKE! are unsupervised methods and they require a pre-defined number of keyphrases to select, we extracted 5, 10, and 15 keyphrases for each corpus and chose the best value for each metric. KeyBART was used in zero-shot settings. The models were evaluated in terms of the full-match F1-score (F1), ROUGE-1 (R1), ROUGE-L (RL) [28], and BERTScore (BS) [50].

The full-match F1-score evaluates the number of exact matches between the original and generated sets of keyphrases. It is calculated as a harmonic mean of precision and recall.

Table 2. Top-10 common keyphrases for the corpora.

Corpus	Keyphrases (keyphrase — number of occurrences)
Krapivin	scheduling – 36, performance evaluation – 25, data mining – 24, computational complexity – 24, parallel algorithms – 22, fault tolerance – 22, approximation algorithms – 22, model checking – 21, distributed systems – 21, preconditioning – 20
Inspec	internet – 199, information resources – 97, probability – 70, computational complexity – 69, optimisation – 60, gender issues – 49, matrix algebra – 47, psychology – 46, human factors – 46, academic libraries – 45
PubMed	children – 24, breast cancer – 21, epidemiology – 20, internet – 19, quality of life – 19, preconception care – 16, pregnancy – 16, apoptosis – 15, cancer – 13, magnetic resonance imaging – 13
NamedKeys	CI – 245, OR – 144, reactive oxygen species – 142, ROS – 134, confidence interval – 131, nitric oxide – 112, NO – 111, HR – 95, ER – 84, oxidative stress – 83
Duc-2001	police brutality – 12, mad cow disease – 11, illegal aliens – 10, Census Bureau – 10, Ben Johnson – 10, Clarence Thomas – 10, investigation – 9, firefighters – 9, welfare reform – 8, crash – 8
KPTimes	U.S. – 1,472, Donald Trump – 1,274, China – 1,122, terrorism – 525, baseball – 510, Russia – 474, elections – 435, Shinzo Abe – 386, football – 364, North Korea – 350

Table 3. A comparison of corpora content proximity, evaluated as in [24]. The value of 1 indicates identical corpora. The higher the score, the greater the difference between corpora.

	Krapivin-A	Krapivin-T	Inspec	PubMed	NamedKeys	DUC-2001	KPTimes
Krapivin-A	1.00	46.24	51.63	141.92	214.31	295.17	267.85
Krapivin-T	46.24	1.00	61.44	125.14	190.75	277.01	238.56
Inspec	51.63	61.44	1.00	98.11	146.79	216.29	202.35
PubMed	141.92	125.14	98.11	1.00	68.72	194.62	121.91
NamedKeys	214.31	190.75	146.79	68.72	1.00	309.29	285.13
DUC-2001	295.17	277.01	216.29	194.62	309.29	1.00	162.96
KPTimes	267.85	238.56	202.35	121.91	285.13	162.96	1.00

The ROUGE-1 score calculates the number of matching unigrams between the model-generated text and the reference. The ROUGE-L score works in a similar way but measures the longest common subsequence. To measure ROUGE-1 and ROUGE-L, the keyphrases for each text were combined into a string with a comma as a separator.

BERTScore utilizes the pre-trained contextual embeddings from BERT-based models and matches words in the source and generated texts using cosine similarity. It has been shown that human judgment correlates with this metric on sentence-level and system-level evaluation. To calculate BERTScore, we use contextual embeddings from RoBERTa-large [29], a modification of BERT that is pre-trained using dynamic masking.

6 Results and Discussion

To answer **RQ1** and **RQ2**, we fine-tuned BART on one corpus and applied it to the other corpora in zero-shot settings. Then we fine-tuned BART on mixed data. For this purpose, we evaluated four strategies:

1. $Domain_{eq}$, fine-tuning the model on the texts of all corpora from one domain (for example, CS domain includes Krapivin-a, Krapivin-T, and Inspec), then testing on each corpus separately. In this strategy, we use an equal number of texts for each corpus. For example, if the size of training sets for Krapivin-A, Krapivin-T, and Inspec are 1,606, 1,605, and 1,400 respectively, we utilize 1,400 random texts from Krapivin-A and Krapivin-T and all texts from Inspec. The overall size of training data is 4,200. The texts from different corpora are mixed in random order.
2. $Domain_{all}$, the strategy is similar to the previous one but we use all texts from each corpus. In this case, the overall size of training data for the example above is 4,611, i.e. 1,606+1,605+1,400.
3. Mix_{eq}, fine-tuning the model on the texts of all corpora using an equal number of texts for each corpus, then testing it on each corpus separately. The texts from different corpora are mixed in random order.
4. Mix_{all}, the strategy is similar to Mix_{eq} but we use all texts from each corpus.

Table 4 shows the performance of baselines on test sets. The best baseline results are underlined. The performance of different methods varies depending on the corpus. For example, KeyBART performs worse on the news domain since this model was pre-trained on scientific texts.

The BART results are presented in Table 5. The results obtained for models fine-tuned on data containing the target corpus are highlighted in blue. Training data are italicized. The scores outperforming baselines are underlined. For mixed training data, we indicate the overall number of training examples in brackets and highlighted in bold the scores that exceed the results of the BART fine-tuned only on the target corpus. The best results among all models (Tables 4 and 5) are marked with an asterisk (*). Table 6 in Appendix A shows a standard deviation for three runs of BART.

Table 4. Baseline results, %.

Target	F1	R1	RL	BS	F1	R1	RL	BS	F1	R1	RL	BS
	KeyBART				*YAKE!*				*TopicRank*			
Krapivin-A	8.58	23.34	19.81	88.26	8.14	20.75	17.58	86.35	6.89	17.68	14.86	87.94
Krapivin-T	5.42	16.61	14.67	87.18	7.09	16.43	14.13	86.14	5.89	15.17	12.48	87.44
Inspec	10.66	29.09	23.55	86.72	13.84	33.80	27.00	86.45	16.32	35.68	25.01	87.38
PubMed	5.70	13.41	12.27	85.56	13.35	20.00	17.28	85.43	11.11	18.61	15.41	86.83
NamedKeys	9.01	21.11	18.30	82.96	20.80	30.62*	22.06*	84.80	19.40	30.55	22.00	84.80
DUC-2001	5.57	11.54	10.37	86.12	13.58	26.93	22.16	85.63	20.88*	30.91*	23.59*	88.51*
KPTimes	4.50	8.52	7.87	83.95	10.05	18.92	16.18	84.83	10.40	14.44	12.74	86.24

The BART fine-tuned on a target corpus outperforms baselines in many cases (Krapivin-A, Inspec, and KPTimes – all metrics; Krapivin-T – F1, R1, and RL; PubMed – R1, RL, and BS; NamedKeys – RL and BS). For DUC-2001, the results of BART are lower than the ones of unsupervised methods, which is probably due to the smaller size of this corpus. The out-of-corpus results are generally lower than the in-corpus ones. For example, when fine-tuning on Inspec (CS domain), the performance in terms of F1 is reduced by 37% and 30% for Krapivin-A and Krapivin-T respectively (both – CS), by 51% and 56% for PubMed and NamedKeys (BM), and by 34% and 78% for DUC-2001 and KPTimes (news). The only exception is the model fine-tuned on Krapivin-A. For Krapivin-T, its results are higher than the in-corpus scores. Thus, fine-tuning on abstracts demonstrated higher scores than the fine-tuning on texts of the papers for the same corpus. The lengths of abstracts and texts were limited to the first 256 tokens due to restrictions on the length of the input sequence and resource limits.

Figure 2 illustrates the effect of adding training examples from other corpora and domains in terms of F1. In our experiments, the effectiveness of the use of additional data varies depending on the characteristics of the corpus. For DUC-2001, which contains few training examples, the use of training examples from other corpora and domains increased the results for all strategies. In contrast, the highest result for KPTimes, which is the largest corpus in our experiments, is obtained using the only target training set. The use of the $Domain_{eq}$ and Mix_{eq} strategies led to a sharp decrease in the size of the training set and the number of targeted examples and negatively affected the model performance. In general, the Mix_{eq} strategy reduces the scores for all corpora except DUC-2001 due to a strong reduction in the amount of training data from the target corpus. Mix_{all} generally improves the performance or at least does not lead to a strong degradation of results[4]. This strategy showed the best results among all models for Krapivin-A (in terms of F1 and BS), Krapivin-T (BS), Inspec (R1, RL, and BS), PubMed (F1, R1, RL), and NamedKeys (RL, BS). Reducing the size of the dataset naturally leads to a decrease in training time. For instance, the training time is 53 minutes 59 seconds for Mix_{all} (21,885 training examples)

[4] This model is available at:
https://huggingface.co/beogradjanka/bart_finetuned_keyphrase_extraction

Table 5. BART results, %.

Target	Training data											
	F1	R1	RL	BS	F1	R1	RL	BS	F1	R1	RL	BS
	Krapivin-A				*Krapivin-T*				*Inspec*			
Krapivin-A	11.77	24.88*	21.46*	88.37	8.38	20.87	17.98	86.86	7.44	21.61	16.94	85.04
Krapivin-T	8.25	18.05	16.05	87.39	7.17	17.25	15.65	87.02	5.05	15.51	13.31	84.30
Inspec	9.85	24.01	19.37	86.63	6.66	19.27	15.97	85.76	20.18*	42.11	35.09	88.42
PubMed	7.97	15.67	13.67	85.74	4.44	11.25	10.12	83.76	6.61	16.98	14.01	83.62
NamedKeys	9.34	16.05	13.63	83.44	5.56	11.87	10.22	81.85	8.89	22.30	16.68	82.58
DUC-2001	6.01	12.84	11.85	86.00	3.19	8.40	7.80	84.34	7.81	17.56	15.48	85.27
KPTimes	3.84	7.17	6.63	83.64	2.46	5.35	4.99	82.30	6.67	11.71	10.25	83.52
	PubMed				*NamedKeys*				*DUC-2001*			
Krapivin-A	6.92	15.67	13.88	86.81	5.22	11.88	10.54	84.81	4.50	14.85	13.05	85.34
Krapivin-T	4.19	11.35	10.33	85.89	2.97	7.77	7.21	83.38	2.65	9.93	9.16	84.14
Inspec	6.08	16.33	13.92	85.46	4.48	11.04	9.51	82.97	4.69	16.22	14.07	83.77
PubMed	13.35	20.96	18.69	86.89*	10.62	17.92	15.43	84.55	6.00	16.14	14.26	85.17
NamedKeys	12.47	20.31	17.13	84.16	20.11	27.04	22.53	85.31	6.42	15.64	13.25	82.75
DUC-2001	5.23	11.64	10.81	85.79	6.27	12.65	11.32	84.61	11.78	24.14	20.65	87.45
KPTimes	6.38	9.89	9.06	84.46	8.94	12.24	10.97	84.22	2.80	8.47	7.78	83.44
	KPTimes				CS_{eq} (4,200)				CS_{all} (4,611)			
Krapivin-A	3.73	7.58	7.20	84.81	12.04	24.35	21.14	88.24	**12.08**	24.49	21.02	88.30
Krapivin-T	2.35	6.27	6.00	84.55	**8.32**	**18.41**	**16.38**	**87.42**	**8.36***	**18.54***	**16.50***	**87.48**
Inspec	3.79	7.80	7.16	83.07	20.04	42.00	34.86	**88.47**	20.01	42.09	34.88	**88.47**
PubMed	8.22	10.77	10.02	85.02	8.11	16.66	14.45	85.57	8.20	16.84	14.55	85.68
NamedKeys	6.31	7.75	7.16	81.76	7.99	16.60	13.75	83.06	7.82	16.52	13.62	83.01
DUC-2001	6.86	14.64	13.15	86.12	4.54	11.36	10.58	85.76	5.16	12.09	11.30	85.87
KPTimes	30.97*	33.98*	28.92*	88.12*	7.11	10.81	9.79	84.51	7.43	10.98	10.05	84.58
	BM_{eq} (1,848)				BM_{all} (3,058)				$News_{eq}$ (432)			
Krapivin-A	6.63	14.89	13.13	86.50	6.30	14.52	12.70	86.11	5.43	13.61	12.05	85.77
Krapivin-T	4.63	11.41	10.52	85.66	4.41	11.20	10.30	85.47	3.34	9.69	8.96	84.89
Inspec	6.11	15.26	13.10	85.10	5.78	14.64	12.36	84.64	5.91	13.74	12.09	83.70
PubMed	13.29	**21.19**	18.48	86.33	13.24	**21.39**	18.46	85.88	7.39	15.80	14.34	85.78
NamedKeys	18.44	25.38	21.21	85.10	**20.69**	**27.77**	**23.13**	**85.42**	7.60	15.03	12.83	83.18
DUC-2001	6.37	13.88	12.48	85.80	6.48	13.53	12.32	85.53	**13.76**	**24.64**	**21.15**	**87.83**
KPTimes	9.43	12.50	11.26	84.83	8.85	12.47	11.21	84.60	4.76	9.34	8.60	84.56
	$News_{all}$ (14,216)				Mix_{eq} (1,512)				Mix_{all} (21,885)			
Krapivin-A	4.68	10.38	9.50	85.61	9.57	21.67	18.61	87.75	**12.52***	24.82	21.41	**88.41***
Krapivin-T	3.41	9.61	8.99	85.58	6.10	15.91	14.27	86.68	**8.24**	**18.09**	**16.19**	**87.50***
Inspec	6.95	15.77	13.81	85.00	13.47	33.34	26.65	87.24	20.00	**42.25***	**35.10***	**88.51***
PubMed	9.71	13.77	12.46	85.60	13.19	20.45	17.85	86.77	**13.71***	**21.89***	**18.94***	86.25
NamedKeys	8.37	11.75	10.43	82.72	13.40	20.36	17.20	84.33	**20.79**	**27.93**	**23.26***	**85.53***
DUC-2001	**12.76**	23.62	20.45	**87.93**	**13.88**	**24.15**	**21.29**	**87.96**	**13.31**	**25.63**	**22.60**	**88.02**
KPTimes	30.53	33.78	28.79	88.10	5.56	9.96	9.18	84.80	30.22	33.49	28.54	88.07

and 3 minutes 59 seconds for Mix_{eq} (1,512 training examples). In this case, the training time decreases by about 20 times using the NVIDIA Tesla T4 GPU.

To answer **RQ3**, we fine-tuned BART on a smaller number of training examples and evaluated the performance by increasing the size of the training data. Similarly to [33], we used the following few-shot transfer procedure. We randomly sampled 50 texts from a target training set, fine-tuned the pre-trained model on this subset, and then tested it on a target test set. Next, we increased the sample size by 50 texts of the target training set and repeated the described procedure, doing so up to 1,000 texts or the end of the training set. We compared the results with the scores obtained using the full target corpus and the out-of-domain corpora mixed in equal proportions. For instance, for Krapivin-A, the mix of out-of-domain corpora includes PubMed, NamedKeys, DUC-2001, and KPTimes. To answer **RQ4**, we first fine-tuned BART on a mixture of out-

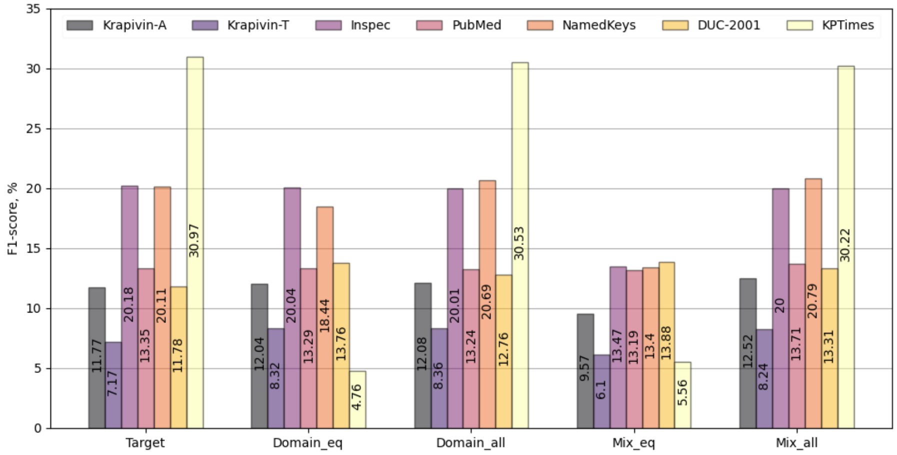

Fig. 2. Adding training examples from other corpora and domains.

of-domain corpora, and then fine-tuned the same model on the texts from the target corpus using the above strategy. We evaluated two options for two-stage fine-tuning. In the first case, we fine-tuned the model on out-of-domain data during half of the epochs (three epochs out of six) and then continued fine-tuning on the target data during the remaining three epochs. In the second case, we doubled the number of epochs and fine-tuned the model within six epochs on both out-of-domain and target data.

The results in terms of F1 are presented in Figure 3. The figure uses the following conventions. *Full target (6 ep)* – fine-tuning on the full target corpus. *Not target_ eq (6 ep)* – fine-tuning on out-of-domain data. *Target (6 ep)* – fine-tuning on a part of the target corpus. *Not target_ eq (3 ep)* → *Target (3 ep)* – fine-tuning on out-of-domain data for three epochs, then fine-tuning on a part of a target corpus for three epochs. *Not target_ eq (6 ep)* → *Target (6 ep)* – fine-tuning on mixed out-of-domain data for six epochs, then fine-tuning on a part of the target corpus for six epochs.

The models with two-stage fine-tuning outperform the ones fine-tuned only on a target corpus on a small target sample size (≈ up to 300 texts). Therefore, the use of out-of-domain corpora allows the use of fewer target data. For some corpora (Krapivin-A, PubMed, and DUC-2001), the models fine-tuned in a two-stage manner outperformed the ones fine-tuned on full target corpora. For Krapivin-A, the F1 outperforming the full target score was obtained using 59% of the target training set. For PubMed and DUC-2001, it took us 43% and 46% respectively. For other corpora with a large full target size, we did not observe the results exceeding the F1 on a full target corpus during this experiment.

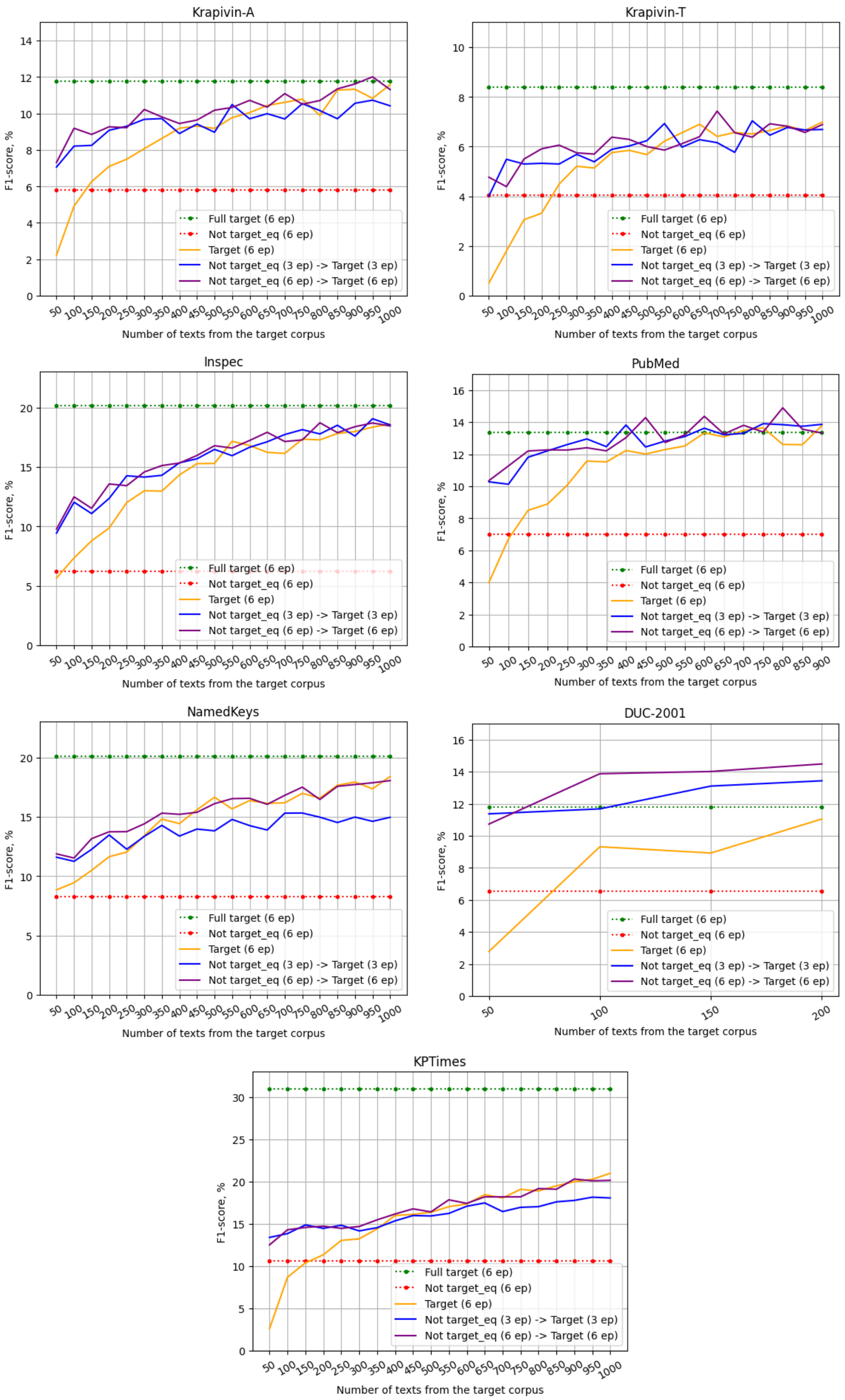

Fig. 3. Performance of BART with and without preliminary fine-tuning on out-of-domain corpora using a varying size of training data.

7 Conclusion

We explored the robustness of the abstractive text summarization models fine-tuned for the task of keyphrase generation. Our experiments are based on BART, a transformer-based denoising autoencoder for pre-training a seq2seq model. We studied the cross-domain limitations of the BART fine-tuned for keyphrase generation across six corpora from three different domains. We also investigated the impact of preliminary out-of-domain fine-tuning to improve the performance of the models under conditions of a small amount of training data. We found that preliminary fine-tuning on out-of-domain data improves the performance of the model in few-shot settings and allows using fewer target data. Future research will focus on transfer learning from a high-resource language, for example, English, to other languages and to Russian in particular.

We explored the robustness of the abstractive text summarization models fine-tuned for the task of keyphrase generation. Our experiments are based on BART, a transformer-based denoising autoencoder for pre-training a seq2seq model. We studied the cross-domain limitations of the BART fine-tuned for keyphrase generation across six corpora from three different domains. We also investigated the impact of preliminary out-of-domain fine-tuning to improve the performance of the models under conditions of a small amount of training data.

We found that preliminary fine-tuning on out-of-domain data improves the performance of the keyphrase generation in few-shot settings and allows the use of fewer target data. Our findings add to a series of results concerning the effectiveness of a two-stage fine-tuning procedure where the transformer-based model is first fine-tuned on the source domain dataset before fine-tuning with the target domain dataset. For instance, similar studies conducted for text classification [34, 47] and named entity recognition [33, 40] have shown that the two-step training procedure can outperform the baseline models fine-tuned only on the target corpus. Our future research will focus on transfer learning from a high-resource language, for example, English, to other languages and to Russian in particular.

A Appendix

Table 6. The values of standard deviation for the BART results.

Target corpus	F1	R1	RL	BS	F1	R1	RL	BS	F1	R1	RL	BS
	Krapivin-A				*Krapivin-T*				*Inspec*			
Krapivin-A	0.36	0.34	0.25	0.05	0.27	0.54	0.43	0.05	0.40	0.32	0.48	0.11
Krapivin-T	0.09	0.16	0.23	0.08	0.18	0.42	0.35	0.05	0.17	0.16	0.03	0.11
Inspec	0.14	0.26	0.18	0.13	0.25	0.41	0.24	0.07	0.21	0.29	0.35	0.07
PubMed	0.27	0.10	0.27	0.08	0.30	0.52	0.38	0.17	0.32	0.30	0.25	0.09
NamedKeys	0.16	0.75	0.55	0.11	0.40	0.80	0.57	0.21	0.13	0.26	0.12	0.07
DUC-2001	0.92	1.36	1.21	0.17	0.26	0.47	0.35	0.08	4.15	7.49	6.35	0.84
KPTimes	0.17	0.23	0.21	0.08	0.25	0.50	0.46	0.16	3.43	5.21	4.29	0.98
	PubMed				*NamedKeys*				*DUC-2001*			
Krapivin-A	0.38	0.48	0.43	0.18	0.12	0.05	0.26	0.03	0.23	0.22	0.33	0.08
Krapivin-T	0.11	0.17	0.18	0.20	0.27	0.29	0.27	0.20	0.06	0.15	0.15	0.06
Inspec	0.31	0.37	0.20	0.22	0.21	0.06	0.16	0.11	0.48	0.88	0.56	0.22
PubMed	0.25	0.43	0.51	0.06	0.39	0.57	0.54	0.22	0.77	0.91	0.64	0.16
NamedKeys	0.32	0.55	0.40	0.07	0.84	1.03	0.77	0.23	0.15	0.16	0.04	0.05
DUC-2001	0.24	0.80	0.75	0.41	1.55	1.44	0.84	0.11	0.52	0.99	0.48	0.16
KPTimes	0.10	0.13	0.10	0.02	0.25	0.28	0.28	0.14	0.14	0.05	0.07	0.12
	KPTimes				CS_{eq}				CS_{all}			
Krapivin-A	0.18	0.28	0.25	0.14	0.27	0.80	0.60	0.06	0.04	0.14	0.13	0.05
Krapivin-T	0.21	0.40	0.35	0.10	0.28	0.43	0.31	0.02	0.22	0.12	0.24	0.11
Inspec	0.28	0.30	0.26	0.16	0.60	0.46	0.46	0.08	0.44	0.41	0.45	0.02
PubMed	0.59	0.71	0.70	0.17	0.48	1.00	0.99	0.24	0.27	0.26	0.45	0.09
NamedKeys	0.12	0.28	0.26	0.09	0.23	0.36	0.28	0.09	0.12	0.54	0.39	0.10
DUC-2001	0.96	0.65	0.70	0.26	0.65	1.32	0.96	0.07	0.43	1.33	0.98	0.28
KPTimes	0.22	0.19	0.21	0.04	0.56	0.70	0.54	0.12	0.25	0.09	0.02	0.05
	BM_{eq}				BM_{all}				$News_{eq}$			
Krapivin-A	0.27	0.30	0.28	0.18	0.14	0.31	0.32	0.13	0.28	0.63	0.55	0.13
Krapivin-T	0.11	0.06	0.23	0.22	0.08	0.16	0.28	0.17	0.08	0.26	0.28	0.09
Inspec	0.09	0.69	0.53	0.14	0.25	0.49	0.46	0.23	0.43	1.24	0.93	0.46
PubMed	0.83	0.38	0.41	0.10	0.48	0.06	0.33	0.10	0.43	0.92	0.75	0.07
NamedKeys	0.52	0.59	0.37	0.13	0.14	0.01	0.22	0.05	0.22	0.45	0.39	0.07
DUC-2001	0.69	2.30	2.30	0.52	0.07	0.62	0.52	0.17	0.41	0.51	1.42	0.19
KPTimes	0.28	0.48	0.35	0.01	0.29	0.11	0.15	0.06	0.05	0.11	0.14	0.05
	$News_{all}$				Mix_{eq}				Mix_{all}			
Krapivin-A	0.56	0.90	0.72	0.25	0.50	0.51	0.68	0.09	0.50	0.52	0.23	0.07
Krapivin-T	0.12	0.58	0.47	0.19	0.36	0.34	0.28	0.21	0.02	0.57	0.42	0.10
Inspec	0.53	1.85	1.28	0.33	0.30	0.20	0.31	0.10	0.33	0.22	0.46	0.02
PubMed	0.15	0.22	0.22	0.10	0.84	0.20	0.11	0.02	0.62	0.35	0.42	0.07
NamedKeys	0.55	0.73	0.52	0.21	0.23	0.38	0.33	0.06	0.21	0.10	0.02	0.08
DUC-2001	0.74	0.27	0.42	0.19	0.54	0.31	0.20	0.08	0.24	0.73	1.09	0.09
KPTimes	0.12	0.13	0.17	0.01	0.16	0.13	0.11	0.01	0.28	0.28	0.17	0.02

References

1. Beliga, S.: Keyword extraction: a review of methods and approaches. University of Rijeka, Department of Informatics, Rijeka **1**(9) (2014)
2. Bird, S.: NLTK: the natural language toolkit. In: Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions. pp. 69–72 (2006)
3. Boudin, F.: PKE: an open source python-based keyphrase extraction toolkit. In: Proceedings of COLING 2016, the 26th international conference on computational linguistics: system demonstrations. pp. 69–73 (2016)

4. Bougouin, A., Boudin, F., Daille, B.: Topicrank: Graph-based topic ranking for keyphrase extraction. In: International joint conference on natural language processing (IJCNLP). pp. 543–551 (2013)
5. Bukhtiyarov, A., Gusev, I.: Advances of transformer-based models for news headline generation. In: Artificial Intelligence and Natural Language: 9th Conference, AINL 2020, Helsinki, Finland, October 7–9, 2020, Proceedings 9. pp. 54–61. Springer (2020)
6. Cachola, I., Lo, K., Cohan, A., Weld, D.S.: TLDR: Extreme summarization of scientific documents. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 4766–4777 (2020)
7. Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., Jatowt, A.: YAKE! keyword extraction from single documents using multiple local features. Information Sciences **509**, 257–289 (2020)
8. Çano, E., Bojar, O.: Keyphrase generation: A text summarization struggle. 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). pp. 666–672 (2019)
9. Çano, E., Bojar, O.: Two huge title and keyword generation corpora of research articles. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 6663–6671 (2020)
10. Chan, H.P., Chen, W., Wang, L., King, I.: Neural keyphrase generation via reinforcement learning with adaptive rewards. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2163–2174 (2019)
11. Chen, W., Chan, H.P., Li, P., King, I.: Exclusive hierarchical decoding for deep keyphrase generation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1095–1105 (2020)
12. Chen, Y.S., Shuai, H.H.: Meta-transfer learning for low-resource abstractive summarization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 12692–12700 (2021)
13. Chen, Y., Song, Q.: News text summarization method based on BART-TextRank model. In: 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). pp. 2005–2010. IEEE (2021)
14. Chowdhury, M.F.M., Rossiello, G., Glass, M., Mihindukulasooriya, N., Gliozzo, A.: Applying a generic sequence-to-sequence model for simple and effective keyphrase generation. arXiv preprint arXiv:2201.05302 (2022)
15. Dung, C.V., et al.: Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction **99**, 52–58 (2019)
16. Gallina, Y., Boudin, F., Daille, B.: KPTimes: A large-scale dataset for keyphrase generation on news documents. In: Proceedings of the 12th International Conference on Natural Language Generation. pp. 130–135 (2019)
17. Gero, Z., Ho, J.C.: Namedkeys: Unsupervised keyphrase extraction for biomedical documents. In: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. pp. 328–337 (2019)
18. Glazkova, A., Morozov, D.: Multi-task fine-tuning for generating keyphrases in a scientific domain. In: 2023 IX International Conference on Information Technology and Nanotechnology (ITNT). pp. 1–5. IEEE (2023)
19. Glazkova, A., Morozov, D.: Applying transformer-based text summarization for keyphrase generation. Lobachevskii Journal of Mathematics **44**(1), 123–136 (2023)
20. Goloviznina, V., Kotelnikov, E.: Automatic summarization of Russian texts: Comparison of extractive and abstractive methods. In: Computational Linguistics and

Intellectual Technologies: Proceedings of the International Conference “Dialogue 2022”. pp. 223–235 (2022)

21. Gupta, S., Gupta, S.K.: Abstractive summarization: An overview of the state of the art. Expert Systems with Applications **121**, 49–65 (2019)
22. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 conference on Empirical methods in natural language processing. pp. 216–223 (2003)
23. Jiang, Y., Meng, R., Huang, Y., Lu, W., Liu, J.: Generating keyphrases for readers: A controllable keyphrase generation framework. Journal of the Association for Information Science and Technology (2023)
24. Kilgarriff, A.: Comparing corpora. International Journal of Corpus Linguistics **6** (11 2001). https://doi.org/10.1075/ijcl.6.1.05kil
25. Krapivin, M., Autaeu, A., Marchese, M.: Large dataset for keyphrases extraction (2009)
26. Kulkarni, M., Mahata, D., Arora, R., Bhowmik, R.: Learning rich representation of keyphrases from text. In: Findings of the Association for Computational Linguistics: NAACL 2022. pp. 891–906 (2022)
27. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: 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. pp. 7871–7880 (2020)
28. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004)
29. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
30. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
31. Malykh, V., Porplenko, D., Tutubalina, E.: Generating sport summaries: A case study for Russian. In: Analysis of Images, Social Networks and Texts: 9th International Conference, AIST 2020, Skolkovo, Moscow, Russia, October 15–16, 2020, Revised Selected Papers 9. pp. 149–161. Springer (2021)
32. Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., Chi, Y.: Deep keyphrase generation. In: ACL 2017-55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers). pp. 582–592 (2017)
33. Miftahutdinov, Z., Alimova, I., Tutubalina, E.: On biomedical named entity recognition: experiments in interlingual transfer for clinical and social media texts. In: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42. pp. 281–288. Springer (2020)
34. Rietzler, A., Stabinger, S., Opitz, P., Engl, S.: Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 4933–4941 (2020)
35. Rubio, A., Martínez, P.: HULAT-UC3M at SimpleText@ CLEF-2022: Scientific text simplification using BART. Proceedings of the Working Notes of CLEF (2022)
36. Schutz, A.T.: Keyphrase extraction from single documents in the open domain exploiting linguistic and statistical methods (2008)
37. Shen, L., Le, X.: An enhanced method on transformer-based model for one2seq keyphrase generation. Electronics **12**(13), 2968 (2023)

38. Song, M., Feng, Y., Jing, L.: A survey on recent advances in keyphrase extraction from pre-trained language models. Findings of the Association for Computational Linguistics: EACL 2023 pp. 2108–2119 (2023)
39. Swaminathan, A., Zhang, H., Mahata, D., Gosangi, R., Shah, R., Stent, A.: A preliminary exploration of GANs for keyphrase generation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 8021–8030 (2020)
40. Syed, M.H., Chung, S.T.: MenuNER: Domain-adapted BERT based NER approach for a domain with limited dataset and its application to food menu domain. Applied Sciences **11**(13), 6007 (2021)
41. Tank, M., Thakkar, P.: Text summarization approaches under transfer learning and domain adaptation settings—a survey. In: Computational Intelligence and Data Analytics: Proceedings of ICCIDA 2022, pp. 73–88. Springer (2022)
42. Vaca, A., Segurado, A., Betancur, D., Jiménez, Á.B.: Extractive and abstractive summarization methods for financial narrative summarization in English, Spanish and Greek. In: Proceedings of the 4th Financial Narrative Processing Workshop@ LREC2022. pp. 59–64 (2022)
43. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: AAAI. vol. 8, pp. 855–860 (2008)
44. Wang, S., Jiang, J., Huang, Y., Wang, Y.: Automatic keyphrase generation by incorporating dual copy mechanisms in sequence-to-sequence learning. In: Proceedings of the 29th International Conference on Computational Linguistics. pp. 2328–2338 (2022)
45. Wright, D., Wadden, D., Lo, K., Kuehl, B., Cohan, A., Augenstein, I., Wang, L.L.: Generating scientific claims for zero-shot scientific fact checking. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2448–2460 (2022)
46. Wu, D., Ahmad, W.U., Chang, K.W.: Pre-trained language models for keyphrase generation: A thorough empirical study. arXiv preprint arXiv:2212.10233 (2022)
47. Yadav, A., Milde, B.: forumBERT: Topic adaptation and classification of contextualized forum comments in German. In: Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021). pp. 193–202 (2021)
48. Ye, H., Wang, L.: Semi-supervised learning for neural keyphrase generation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 4142–4153 (2018)
49. Zhang, J., Zhao, Y., Saleh, M., Liu, P.J.: PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In: Proceedings of the 37th International Conference on Machine Learning. pp. 11328–11339 (2020)
50. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating text generation with BERT. In: International Conference on Learning Representations
51. Zmandar, N., El-Haj, M., Rayson, P.: A comparative study of evaluation metrics for long-document financial narrative summarization with transformers. In: International Conference on Applications of Natural Language to Information Systems. pp. 391–403. Springer (2023)
52. Zolotareva, E., Tashu, T.M., Horváth, T.: Abstractive text summarization using transfer learning. In: ITAT. pp. 75–80 (2020)

Thank you to the reviewers and to the editors for the constructive feedback allowing us to further improve the manuscript. We believe that this revised manuscript addresses the reviewers' concerns. We look forward to working with you to further improve the current version, if necessary.

SUBMISSION: 28
TITLE: Cross-Domain Robustness of Transformer-based Keyphrase Generation

---------------------- REVIEW 1 --------------------
SUBMISSION: 28
TITLE: Cross-Domain Robustness of Transformer-based Keyphrase Generation
AUTHORS: Anna Glazkova and Dmitry Morozov

----------- Overall evaluation -----------
SCORE: 1 (weak accept)
----- TEXT:
The paper considers efficiency of model transfer in the keyword extraction task.
The BART language model is trained on one dataset then applied to another dataset.
The author found that preliminary fine-tuning on out-of-domain data improves the performance of the model in few-shot settings and allows using fewer target data.

Comments

1) The paper does not contain the description of related work studied the transfer for summarization and keyword extraction tasks;

Thank you for this comment. We've added the Related Work section (pages 3-4) containing a brief review of the studies in the fields of abstractive text summarization using pre-trained transformers and keyword selection.

2) Measures of performance are not explained. I can guess what R1 and RL mean, but I cannot guess what is BS.

Thank you for this comment. The interpretation for this abbreviation (BERTScore) is given in Section 5, in the first paragraph.

3) No analysis is presented, which compare the model transfer results in keyword extraction with similar studies for other tasks.

Thank you for this comment. We made corrections to the conclusion of the paper. We indicated several works containing similar results for a two-stage fine-tuning procedure in two NLP domains.

To conclude:
The paper contains a lot of experiments.
But it seems that the behavour of models is predictable, similar results were shown for other tasks.

---------------------- REVIEW 2 --------------------
SUBMISSION: 28

TITLE: Cross-Domain Robustness of Transformer-based Keyphrase Generation
AUTHORS: Anna Glazkova and Dmitry Morozov

----------- Overall evaluation -----------
SCORE: 2 (accept)
----- TEXT:
In this paper authors present a large number of experiments related to deep learning techniques for keyphrase generation. The general idea is to train a model on some dataset and see how it will perform on another one, possibly belonging to a different domain. For this, 7 different datasets, grouped into 3 domains are used. Furthermore, authors proposed and evaluated 4 strategies for fine-tuning on a mixed data.

The paper is clearly written (explicitly stated RQs help a lot), discusses state-of-the-art methods, and is relevant to the community.

Some comments:
1) "The use of the Domaineq and Mixeq strategies led to a sharp decrease in the size of the training set". It would be interesting to look at the training time reduction in this case.

Thank you for this comment. Reducing the size of the dataset naturally leads to a decrease in training time. For instance, the training time is 53 minutes 59 seconds for Mix_all (21,885 training examples) and 3 minutes 59 seconds for Mix_eq (1,512 training examples). In this case, the training time decreases by about 20 times using the NVIDIA Tesla T4 GPU. We specified this issue in the Results section.

2) It would be better to have dedicated discussion section where results for all RQs are summarized (without mixing-in details on what was done for a particular RQ).

Thank you for this comment. In the previous version of the manuscript, the experimental setup and the results were presented in the same section. In the current version, we made a separate section for the experimental setup (Section 5). The resulting tables and their discussion are placed in Section 6.

3) There are numerous issues with articles. I suggest doing manual proofreading or using ChatGPT, which is really good for this task.

Thank you for this comment. We carefully read the text and corrected the issues you mentioned below.

Misc:
1) language, style, layout
> The most unsupervised approaches for keyphrase selection
Just "Most..." or "The most common/famous/... unsupervised"
> deep-learning
deep learning
> However, like other fine-tuned models, it probably shows lower performance
Conversational style
> CS - computer science, BM - biomedical, A - abstract, and T - text (body).
em dash needed (---)
> YAKE! [4]
layout flaw. can be fixed with: YAKE~\cite{...}
> scheduling – 36
em dash needed in the whole table, and in many other places
> Nltk, Pke, ...

Use {NLTK} in bibtex entry to generate capitalized name.
2) Captions for all tables should be above the table, not below as we use LNCS format. In fact, only ACL conferences place captions below.
3) In tables, after the "." there should be the same number of digits inside any fixed column (do not omit zero)
Thank you/ We've fixed these issues.
----------------------- REVIEW 3 ---------------------
SUBMISSION: 28
TITLE: Cross-Domain Robustness of Transformer-based Keyphrase Generation
AUTHORS: Anna Glazkova and Dmitry Morozov

----------- Overall evaluation -----------
SCORE: 2 (accept)
----- TEXT:
The paper is devoted to an important problem of keyphrase generation, which can be used, for example, in patent or scientific information retrieval systems. The paper contains a clear description of the goals and contributions. Training data and results are also well described.

The drawbacks are the following.
1. The paper misses the "Related work" section. It is recommended to add one. Although the Introduction has a discussion of some studies, it is really shallow and should be extended.

Thank you for this comment. We added the Related Work section.

2. Some non-BART-based approaches could be added as baselines.

Thank you for this comment. We compared the results with two non-BART-based baselines, such as TopicRank and YAKE. The baseline results are presented in Table 4.

3. The paper lacks the parameters of BART fine-tuning (loss, optimizer, epochs, etc.).

Thank you for this comment. We specified the implementation details in Section 4 (first paragraph).

----------------------- REVIEW 4 ---------------------
SUBMISSION: 28
TITLE: Cross-Domain Robustness of Transformer-based Keyphrase Generation
AUTHORS: Anna Glazkova and Dmitry Morozov

----------- Overall evaluation -----------
SCORE: 1 (weak accept)
----- TEXT:
The article is devoted to the study of machine learning methods for solving the problem of describing the content of the text by keywords by the authors. The authors consider this task as a task of annotating with keywords. More precisely, the authors explore the problems of transfering an algorithm trained on one collection to another.

From a formal point of view of the application of machine learning techniques, it would seem that everything is done correctly. At the same time, the comparison results are very low. The authors do not give a reasonable interpretation of why this is so.

I see the main problem of this study in that the authors do not fully understand the problem they are solving.
The assignment of keywords to the document has several purposes: (1) indeed, summarizing the content in a concise form, so that the reader can understand the meaning of the article when "quickly" browsing the headings (in this case, according to the bibliographic tradition, it is necessary to choose a small number of "orthogonal in meaning" keywords); (2) the appearance of an article in the search results for experts in the subject area in the collection of articles; (3) and as a development of the second alternative - in fact, the classification of the article according to some "classifier" accepted in a narrow subject area based on widely (frequently) used well-known terms.
That is, different authors may use different keyword selection algorithms.

In fact, these goals contradict each other to a certain extent. At the same time, the significance of a particular goal largely depends on the organization of documents in a particular collection. For example, for collections of images to serve the second purpose, in addition to a short set of keywords (5-10), a set of a large number of keywords (100) is also often used.
The authors of scientific texts, due to the peculiarities of the organization of archives of scientific publications, often focus on the third goal. In fact, there is a choice of "headings" characteristic of works on similar topics. To make it easier for the user to find "similar" publications.

Therefore, it is not complete to consider the solution of the problem of assigning keywords only as an summarization task. Perhaps it makes sense to try to consider this problem also as a classification task for similar documents, when the most frequent keywords take precedence in the selection.

Moreover, transferring from one collection to another may not work precisely because of the different traditions of using keywords specific to different collections.

Thank you for the comprehensive review of our work and for your comment. Indeed, the task of selecting keywords is understood by researchers in different ways. As provided in the surveys of S. Beliga [1] and Cano and Bojar [8], keyword selection approaches can be roughly divided into three categories: i) actual keyword extraction, ii) keyword assignment, and iii) keyphrase generation. In this work, we used the third approach. However, we also used classic approaches to extracting keywords as baselines. In the Related Work section, we also report the variety of approaches.