Keyphrase Generation by Utilizing BART Finetuning and BERT-Based Ranking

Diya A[†] Mizuho IWAIHARA [‡]

Graduate School of Information, Production and Systems, Waseda University

2-7 Hibikino, Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0135 Japan

E-mail: †adiya@moegi.waseda.jp, ‡ iwaihara@waseda.jp

Abstract Keyphrases are condensed summaries of text information, representing a document's theme and core idea. Keyphrase extraction approaches based on statistics and deep learning have received widespread attention. The extractive approaches have obvious limitations like keyphrases must occur in the document. However, a considerable amount of human-annotated keyphrases are absent keyphrases such that they are not occurring in the source document explicitly, which cannot be extracted by the extractive methods. Recent keyphrase generation is based on sequence-to-sequence deep learning models, generating present and absent keyphrases from the semantic representation of the document. However, recall and precision of generated absent keyphrases still need to be improved. We propose a two-step keyphrase generation model called Ranking-KPG to solve these problems. Ranking-KPG consists of a finetuned BART-based keyphrase generator "KeyBART", coupled with a BERT cross encoder which ranks generated keyphrases. Our model can generate absent keyphrases and shows superior results over strong baselines in four widely used datasets.

Keyword Natural language processing, keyphrase generation, fine-tuning, generative language model, sequence generation

1. Introduction

A keyphrase is a phrase composed of one or several words, which can highly summarize the document's central idea. Keyphrases provide assistance for efficient use of text resources and play an essential role in the fields of natural language processing such as document classification and information retrieval. The task of extracting keyphrases from the content of a document is called keyphrase extraction. The traditional keyphrase extraction methods are mainly completed by acquiring a list of keyphrase candidates, then ranking candidates on their importance to the source text [1].

Keyphrases of a document often do not appear directly in the document, which is called absent keyphrases. Obviously, keyphrase extraction methods cannot be applied for absent keyphrase prediction. For example, when people see phrases such as "rouge-score" in the document, they will naturally mark the paper with keyphrases such as "natural language processing" and "document summarization," no matter they are occuring or not. Extractive methods cannot produce absent keyphrases. Generation of keyphrases is more challenging than extraction since candidate phrases are not restricted to the document or the corpus. Up to now, CopyRNN [7], CatSeq [9], and its variants are the only known supervised keyphrase generation models. All of the models are based on a recurrent generative model that can be used to generate multiple keyphrases as delimiter-separated sequences. This type of models is capable of generating diverse keyphrases, where the number of outputs is controlled.

However, generation performance of Catseq for absent keyphrases is still limited. In this paper, we propose a new keyphrase generation model which utilizes the pretrained language model BART [5]. As a pre-trained seq2seq model, BART shows excellent text summarization performance and achieves new state-of-the-art results on text generation tasks. The keyphrase generation task is related to document summarization. BART's bidirectional encoder and autoregressive decoder are expected to be transformed for keyphrase generation through finetuning.

In terms of present keyphrase extraction, traditional ranking-based models still have obvious advantages over generative models. Due to this observation, we also utilize BK-Rank [6] in addition to the finetuned BART model for final keyphrase selection. BK-Rank first selects candidate phrases based on POS tagging and then gives a relevance score of candidate phrases with the original document for ranking, which shows outstanding extraction performance on multiple datasets. Our model Ranking-KPG combines the outputs of the finetuned

BART and BK-Rank models, and ranks the combined results through another finetuned BERT cross-encoder, for selecting the top-k prediction as to the final output.

Our absent keyphrase evaluation results show that, compared with Catseq-based generative models, our BART-based models show superior results in F1@10 and F1@O metrics on the four widely used test datasets.

The main contributions of this paper are as follows:

1. Proposing KeyBART, a new keyphrase generation approach based on a generative language model, enhancing the performance of absent keyphrase generation to more than triple the performance of CatSeq on all datasets.
2. Proposing Ranking-KPG, which combines the keyphrase generation and keyphrase extraction models by utilizing BERT cross-encoder for ranking, achieving a higher recall rate on present keyphrases and further enhancing the advantage of BART-based model on absent keyphrases.

2. Related Work

Keyphrase generation is an emerging research topic, which is highly related to the two areas of traditional keyphrase extraction and text summarization.

2.1 Keyphrase Extraction & Generation

In the past few decades, traditional keyphrase extraction methods have been extensively studied. Most of them follow two steps for extraction. First, linguistic features such as part-of-speech tags are used to determine a list of phrase candidates. Second, a ranking algorithm is adopted to rank the candidate list, and the top-ranked candidates are selected as keyphrases. A wide variety of methods were applied for ranking, such as bagged decision trees, multi-layer perceptron, support vector machine, and PageRank. Subramanian uses pointer networks to point to the start and end positions of keyphrases in a source text.

To solve the problem that the keyphrase extractions method cannot produce absent keyphrases. Meng et al. first proposed CopyRNN, a neural model that both generates words from vocabulary and points to phrases from the source text [7]. Based on the Copy-RNN architecture, Chen et al. [2] introduce the idea of reinforcement learning into the keyphrase generation task.

2.2 BART

BART is a denoising autoencoder for the pre-training sequence-to-sequence model. It first destroys the text by using an arbitrary noise function and then rebuilds the original text for training through the learning model [5]. BART uses a transformer-based machine translation structure. From the structural point of view, it is like a combination of BERT (with a bi-directional encoder) and GPT (with a left-to-right decoder). Due to the autoregressive decoder, BART can directly perform sequence generation tasks by finetuning, such as dialogue and summary. When compared with the extractive document summarization models, BART's performance on Xsum (a highly abstract dataset) is far better than the previous model based on BERT. Also, the sample quality has been significantly improved.

2.3 BK-Rank & Cross-Encoder

We utilize BK-Rank [6] as a supervised keyphrase extraction model. It first selects candidate phrases from the document through part-of-speech (POS) tagging. Then a cross-encoder [3] computes self-attention between the original document and the candidate phrase, to capture the relationship between these two parts. Finally, the candidate phrases are ranked by their relevance scores and the top-N phrases are selected as final outputs. In addition, in response to the problem of output diversity, MMR (Maximal Marginal Relevance) is introduced to reduce the resulting redundancy.

For text similarity measures, a bi-encoder generates embeddings for each document. Then similarity between two documents is efficiently calculated by the cosine similarity between them. On the other hand, a cross encoder-based similarity is based on the self-attention of the Transformer over the concatenation of two documents, capturing more relatedness between the two documents than the bi-encoder.

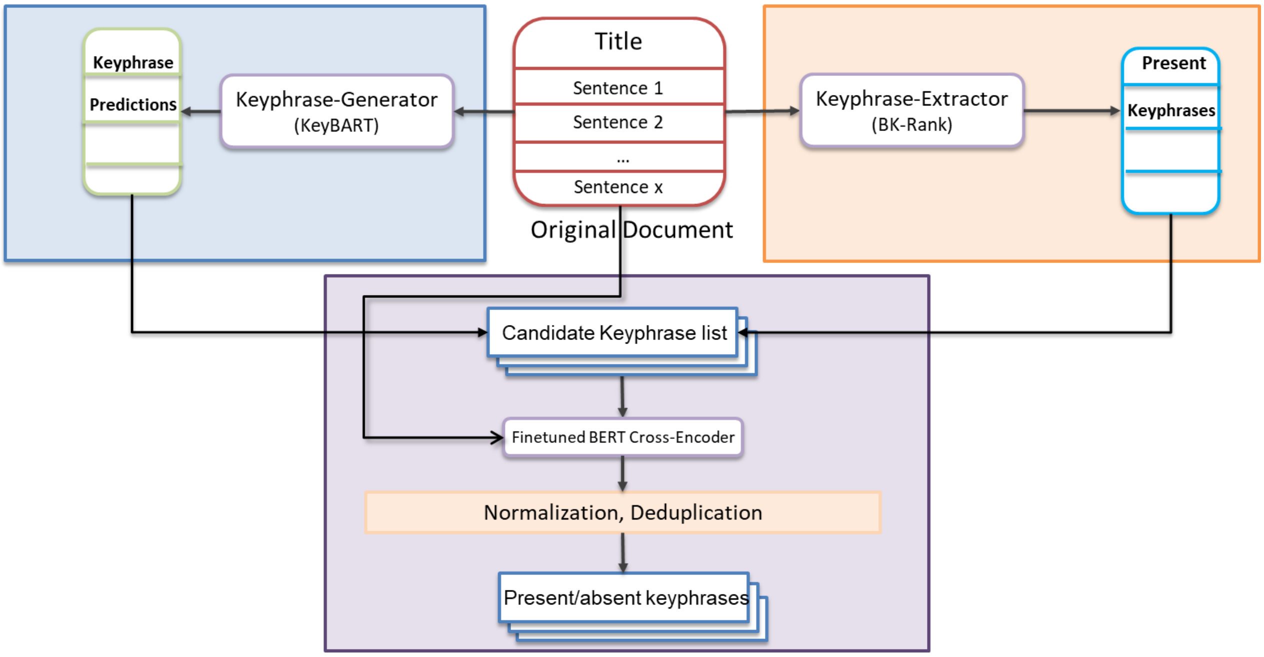


Figure 1. Overall structure of our proposed model

3. Methodology

Figure 1 shows an overview of our proposed model. The main process of the model is as follows. First, we utilize the extractive model BK-Rank to extract present keyphrases from the target document. Then keyphrase generation is performed on the same document by the keyphrase-focused BART model KeyBART. Then the resulting keyphrases of BK-Rank and KeyBART are merged, normalized, and duplicate phrases are removed. Finally, the merged list and target document are entered into a finetuned BERT cross-encoder. The phrase list is sorted according to the relevance score produced by the BERT cross-encoder. Finally, the top-k candidate phrases containing both present and absent keyphrases are returned.

For keyphrase-focused BART finetuning, we choose to use the "Facebook/Bart-Large" model as the pre-trained model, in which both encoder and decoder have 12 layers. We use the KP20K training dataset [11] for finetuning. Our finetuned BART-based keyphrase generator KeyBART's structure is shown in Figure 2.

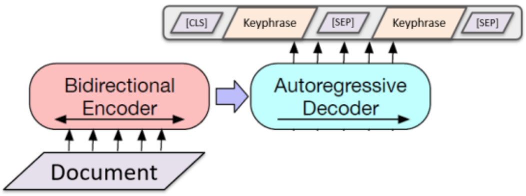


Figure 2. Structure of KeyBART

We combine each document and their corresponding keyphrases as training samples for BART to be able to generate keyphrases. First, we select 'src' (source text) and 'id' (document id) from the source file. Then we select the corresponding 'tgt' (target, ground-truth keyphrases) from the target file. A new dataset dictionary item consisting of selected 'src', 'id', 'tgt' is generated. The above operation is repeated for each data in the training set. Finally, all dictionary items are combined as a new dataset for BART finetuning. 513918 samples are obtained in total. During the BART model training, we set the max input length and max target length to 800 and 256 tokens, respectively. The learning rate is set to 2e-5, and the label smoothed cross-entropy loss is used as the loss function. The training batch size of BART-Large is set to 4, and training epochs are set to 5.

The advantage of KeyBART is also reflected in the generation diversity. For keyphrase extraction and key phrase generation tasks, previous ranking-based methods tend to generate many redundant candidate phrases with extremely close meanings. In the BART generation process, we can control the diversity of the generated phrases by adjusting the "diversity_penalty" parameter. But even without this, there are fewer redundant phrases that occur in KeyBART's output.

3.1 Cross-encoder Training

For the cross-encoder training, we again choose to use KP20K because of the vast amount of samples. To train the cross-encoder, we also need to prepare negative samples. For each document, we use the raw text, a positive sample, and a negative sample to build a training sample. The positive sample is one of the ground-truth keyphrases, where both present and absent keyphrases are included. Negative samples are taken from the false-positive predictions from KeyBART's output, which means wrong keyphrase predictions from KeyBART's output. We use Bert-base-uncased as a pretrained model, with batch size 16, and training for 2 epochs. Our finetuned cross-encoder's structure is shown in Figure 3.

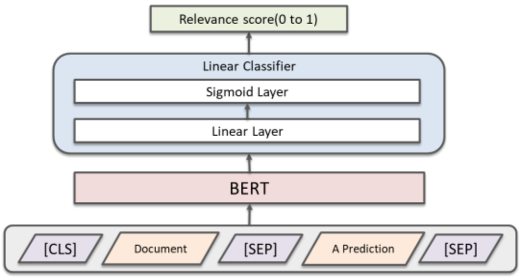


Figure 3. Scoring keyphrases by BERT Cross-encoder

3.2 Evaluation system

Here, we describe post processing on keyphrase prediction results, before computing evaluation scores.

I. Inputs are the keyphrase prediction list from outputs of the keyphrase generation method, ground-truth keyphrases list, and the target document.

II. Remove null prediction results, such as '.' ',' and <unk> marks) from the prediction list.

III. Stem the elements in the prediction list and ground-truth list to eliminate the influence of personal and tense.

IV. Compare the prediction results with ground-truth one by one, and judge whether the phrase has appeared in the original document. If it is true, the keyphrase is marked as present, otherwise absent.

V. Calculate P / R / F1 scores for present and absent keyphrases, respectively.

3.3 Evaluation Measures

We design two evaluation measures named measure (a) and measure (b). Their differences are mainly derived from the fact that reference (ground-truth) keyphrases of documents are a mix of present and absent keyphrases, and the ratio between the present and absent keyphrases are varying. Existinc literature work on keyphrase generation measures precision/recall F1 scores separately on the two lists of absent and present keyphrases. However, we believe that the keyphrase generation system needs to select final keyphrases from both absent and present keyphrases. Indeed, the output of all the generative models is only one list containing both present and abstract keyphrases. Before output predictions, the model will sort all the candidates based on the beam score generated by the beam search [12] process, and produce the most plausible keyphrase prediction.

For the definition of "Top-k predictions" used for evaluation, Meng et al. adopt the method of dividing the prediction list into two independent lists of present and absent predictions and extracting the top-k predictions respectively to the evaluation system. We also adopt this evaluation measure as the measure (a). Figure 4 illustrates measure (a), where red background indicates the present keyphrases and blue for absent ones. The numbers indicate the order of grouped predictions. By analyzing the output of the generative model, we observe that, since absent keyphrases are more difficult to be predicted, the first correct absent prediction often occurs after more than 5 or more predictions. But we still evaluate top-k by using the evaluation measure (a).

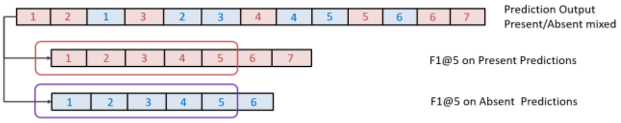


Figure 4. The process of evaluating the top-5 predictions using measure (a).

We believe that measure (a) could not show the accuracy of the entire model well, and present and absent should have the same priority in the evaluation step. Therefore, we propose measure (b) for evaluation, as shown in Figure 5. In this measure, we simply extract the top-k predictions from the output list and divide them into two parts by present and absent keyphrases. These two parts will be evaluated separately.

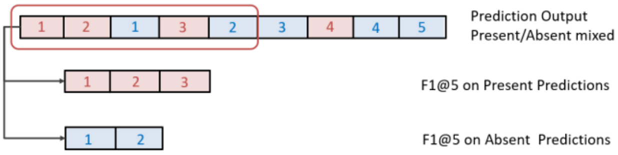


Figure 5. The process of evaluating the top-5 predictions using **measure (b)**.

4. Datasets and Experiments

For model training, Meng et al. proposed the KP20K dataset. KP20K is based on major online libraries (ACM digital library, ScienceDirect, Wiley, etc.), containing 567,830 scientific papers and corresponding keyphrases designated by the authors. In addition, they divided the 527,830 documents for model training, 20,000 documents for verification, and another 20,000 documents as the test set to evaluate the model performance.

We select four currently widely used keyphrase datasets to evaluate the performance of our models and baseline methods. Inspec contains the abstracts of 2,000 documents. We sampled 500 documents as the test set. Krapivin provides the full text of 2,304 documents and keyphrases formulated by the authors. We selected the first 400 papers in alphabetical order as test cases. We use all 211 documents in the NUS dataset and these corresponding keywords as test cases. We also sampled 100 documents (scientific research papers from ACM Digital Library) from the SemEval-2010 dataset.

4.1 Baseline Models

We chose four widely used methods in keyphrase extraction and generation as the baseline models to participate in the comparison.

TF/IDF is a statistical method used for information retrieval and mining. This technique is used to evaluate the importance of a word or phrase to a document in a corpus.

PageRank is a link analysis algorithm designed to solve the problem of web page relevance ranking. TextRank draws on the idea of PageRank, constructs a network through the adjacent relationship between words, and then iteratively calculates the rank value of each node to build a candidate keyphrase word graph. Finally, top-ranked phrases are returned.

CatSeq is a recent well-known keyphrase generation model, based on a recurrent generative model that generates multiple keyphrases as delimiter-separated sequences [9]. CatSeqD is a variant of CatSeq, which refers to the model augmented with orthogonal regularization and semantic coverage mechanism [9]. CatSeq2RF1 refers to the CatSeq-based model trained by the reinforcement learning approach [2].

4.2 Experimental Setting

As evaluation metrics, we choose the commonly used precision, recall, and F1-score as the primary metrics and utilize two arranged metrics to evaluate the quality of the model generation.

$$\text{Precision}:\ P@k=\frac{|\hat{y}_{:k}\cap y|}{|\hat{y}_{:k}|}$$

$$\text{Recall}:\ R@k=\frac{|\hat{y}_{:k}\cap y|}{|y|}$$

$$\text{F1 score}:\ F_1@k=\frac{2*P@k*R@k}{P@k+R@k}$$

Here, y represents ground-truth keyphrases for the given source text. $\hat{y}$ represents a list of unique keyphrases ordered by the quality of the predictions. $\hat{y}_{:k}$ represents only the top-k predictions are used for evaluation. Measure (a) uses top-k present (or absent) predictions for $\hat{y}_{:k}$, while measure (b) uses top-k preditions where present and absent keyphrases are combined, for $\hat{y}_{:k}$.

Previous studies show that the number of keyphrases in different documents is diverse, so it is sometimes unfair to use the traditional fixed-rank metrics such as F1@K for evaluation. Following Yuan's experiment setting [7], we also evaluate by two variable-length evaluation metrics, F1@O and F1@M. O denotes the number of oracles (ground truth) keyphrases, which means that for each data example, the number of predicted phrases taken for evaluation is the same as the number of ground truth keyphrases. M denotes the number of predicted keyphrases. In this case, simply all the predicted phrases are taken for evaluation without truncation.

5. Experimental Results

We test the performance of the proposed model on four widely used datasets and evaluate concerning present and absent keyphrases.

5.1 Results on Evaluation Measure (a)

We first evaluate the generated results using evaluation measure (a), where the top-k predictions are selected by presence. It means that all the predictions will be first divided into two lists on present and absent. Then the top-k predictions will be extracted from each list for scoring.

In terms of present keyphrases, we show F@10 and F1@O scores for comparison with the traditional methods. We also show the recall rate "R@M" to measure the maximum extraction performance.

Table 1 shows the performance comparison between our proposed models and the baseline models. For the

Datasets	Inspec			Krapivin			NUS			Semeval		
	F1@10	F1@O	R@M	F1@10	F1@O	R@M	F1@10	F1@O	R@M	F1@10	F1@O	R@M
TF/IDF	18.32	20.01	-	9.19	10.52	-	12.90	14.12	-	11.54	12.32	-
TextRank	34.78	33.23	-	20.15	18.20	-	28.97	30.21	-	24.07	25.15	-
CopyRNN	30.09	31.58	48.21	31.04	34.32	54.99	36.41	**42.50**	51.20	30.92	31.14	47.32
CatSeq	39.48	39.67	50.43	33.53	33.26	60.33	35.67	41.14	54.07	**32.25**	**32.87**	46.92
CatSeqD	33.40	32.61	40.34	28.46	32.36	55.50	**36.94**	38.05	49.21	32.22	32.28	39.22
Catseq-2RF1*	38.02	40.18	-	31.42	**34.55**	-	35.92	40.29	-	31.42	30.14	-
BK-Rank	35.58	35.15	70.67	23.04	22.66	54.16	28.61	28.22	53.84	22.32	22.88	60.43
New Approaches												
KeyBART	**41.16**	**41.09**	44.44	**33.77**	30.71	52.17	36.58	40.77	44.68	30.98	32.73	37.99
Ranking-KPG	26.13	25.35	**79.98**	17.33	13.00	**70.32**	19.86	20.49	**64.66**	18.45	23.47	**70.47**

Table 1. Present keyphrase prediction performance on scientific paper datasets by evaluation measure (a). The best results are shown in bold. Model names with * represent its result is computed from released keyphrase predictions.

present keyphrases, BART, Catseq, and CopyRNN have their advantages on different datasets, but the gap is quite limited. Also, for present keyphrases, when taking leading 50 output from BK-Rank, It shows that performance of f1@k is falling behind of CatSeq, but can realize higher recall rate. BK-Rank's results are utilized by Ranking-KPG, the maximum recall rate of Ranking-KPG is appraently higher than the other models on absent keyphrases. But the cross-encoder ranking mechanism still performs not so well currently on present keyphrases. Due to this, F1@10 and F1@O are not as good as the other methods. Therefore, our cross-encoder-based ranking mechanism still has great potential for improvement.

Table 2 shows the performance comparison between our new models and the baseline models on absent keyphrase generation, where measure (a) is used. We show F1@5, F1@10, and F1@O scores for comparison with CatSeq and Its varients. The experimental results in Table 2 show that for absent keyphrases generation, KeyBART shows an overwhelming advantage over the baseline models. The F1 score of the model based on Finetuned BART-large is close to three times or above than CatSeq. Also higher than CatSeqD on all metrics. Since evaluation measure (a) treats present and absent keyphrases separately, our ranking mechanism does not have impact on internal ranking for absent keyphrases, and the results are similar with just using KeyBART.

5.2 Results by Evaluation Measure (b)

In this experiment, we use measure (b) to evaluate the generated results. This measure evaluates the top-k predictions in the combined list of present and absent predictions, which reflects the weighting between present and absent keyphrases of the ground truth.

Table 3 shows the performance comparison for present keyphrase generation when using measure (b). The F-scores of all measures drop slightly. In this case, the gap between the measures has barely changed compared with the result of measure (a). This is because present keyphrases are easier to obtain a higher beam score, so the top-k predictions extracted by measures (a) and (b) have large overlapping parts.

Datasets	Inspec			Krapivin			NUS			Semeval		
	F1@5	F1@10	F1@O	F1@5	F1@10	F1@O	F1@5	F1@10	F1@O	F1@5	F1@10	F1@O
CatSeq	1.58	1.38	1.47	0.70	0.51	0.66	0.51	0.58	0.55	1.17	0.90	0.81
CatSeqD	1.27	0.97	0.94	4.20	3.71	4.23	4.34	3.78	3.85	3.66	3.61	3.08
Catseq-2RF1*	1.92	1.45	1.87	0.93	0.47	0.79	0.67	0.81	0.79	1.35	1.24	1.01
New Approaches												
KeyBART	3.03	3.11	3.05	4.91	4.78	5.11	5.19	5.15	5.24	4.07	4.00	3.99
Ranking-KPG	**3.03**	**3.11**	**3.05**	**4.91**	**4.78**	**5.11**	**5.19**	**5.15**	**5.24**	**4.07**	**4.00**	**3.99**

Table 2. Absent Keyphrase prediction performance on scientific paper datasets of evaluation measure (a). The best results are shown in bold.

Table 4 shows the performance comparison for absent keyphrase generation, where measure (b) is used. In this case, our KeyBART leads by 2 to 20 times than Catseq in all datasets except Inspec, while keeping similar trends with present keyphrase performance. Also stronger than CatSeqD in all the tests. In the Krapivin and NUS datasets, all the top 5 predictions of the original Catseq did not contain any true positive absent keyphrase predictions. The addition of the BK-Rank and cross-encoder ranking mechanism further expands this advantage, so that Ranking-KPG is achieving the best results in any indicators, demonstrating excellent performance on absent keyphrases generation.

Datasets	Inspec			Krapivin			NUS			Semeval		
	F1@10	F1@O	R@M	F1@10	F1@O	R@M	F1@10	F1@O	R@M	F1@10	F1@O	R@M
TF/IDF	18.32	20.01	-	9.19	10.52	-	12.90	14.12	-	11.54	12.32	-
TextRank	34.78	33.23	-	20.15	18.20	-	28.97	30.21	-	24.07	25.15	-
CopyRNN	28.25	29.95	48.21	27.33	32.46	54.99	33.10	**36.89**	51.20	29.42	30.68	47.32
CatSeq	38.45	38.98	50.43	26.46	**33.36**	60.33	34.78	34.54	54.07	32.76	**31.64**	46.92
CatSeqD	30.13	29.55	40.34	27.12	30.13	55.50	**35.21**	36.29	49.21	30.64	31.09	39.22
Catseq-2RF1*	37.39	38.94	-	24.33	30.24	-	33.40	31.05	-	**33.95**	32.78	-
BK-Rank	35.38	35.15	70.67	23.04	22.66	54.16	28.61	28.22	53.84	22.32	22.88	60.43
New Approaches												
KeyBART	**39.52**	**39.43**	44.44	**27.62**	31.99	52.17	33.76	36.38	44.68	30.60	30.98	37.99
Ranking-KPG	17.05	17.89	**79.98**	12.22	14.76	**70.32**	13.76	16.63	**64.66**	13.09	15.55	**70.47**

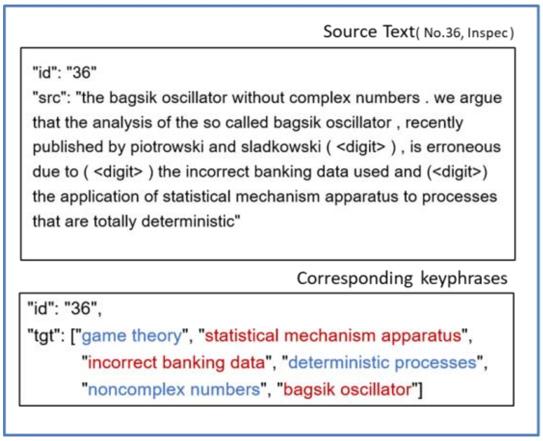
Table 3. Present keyphrase prediction performance on scientific paper datasets of evaluation measure (b).

Datasets	Inspec			Krapivin			NUS			Semeval		
	F1@5	F1@10	F1@O	F1@5	F1@10	F1@O	F1@5	F1@10	F1@O	F1@5	F1@10	F1@O
CatSeq	0.24	0.84	0.76	0.00	0.32	0.11	0.00	0.14	0.19	0.18	0.54	0.81
CatSeqD	0.39	0.58	0.63	1.48	2.86	2.06	1.46	2.86	2.70	0.92	1.97	2.50
Catseq-2RF1*	0.36	0.97	0.88	0.03	0.41	0.26	0.00	0.17	0.24	0.23	0.47	1.09
New Approaches												
KeyBART	1.29	2.72	2.62	3.41	4.32	3.39	2.90	3.88	3.01	3.43	3.86	**3.99**
Ranking-KPG	**2.54**	**3.06**	**3.05**	**4.02**	**4.78**	**4.08**	**3.89**	**4.55**	**4.45**	**3.79**	**4.00**	**3.99**

Table 4. Absent Keyphrase prediction performance on scientific paper datasets of evaluation measure (b). The best results are shown in bold.

6. Example of Keyphrase Generation

To illustrate the process of keyphrase generation and performance evaluation, we show a generation example for further reference. Figure 6 shows the result of a random article (No.36 of the Inspec test dataset). This article contains an abstract of a scientific paper, corresponding to six designated keyphrases, of which three present keyphrases and three absent keyphrases.



Source Text(No.36, Inspec)

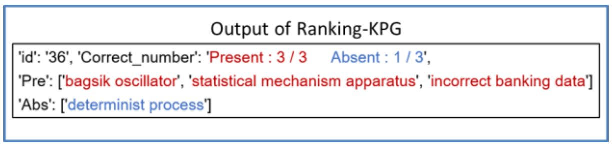
```
"id": "36"
"src": "the bagsik oscillator without complex numbers . we argue
that the analysis of the so called bagsik oscillator , recently
published by piotrowski and sladkowski ( <digit> ) , is erroneous
due to ( <digit> ) the incorrect banking data used and (<digit>)
the application of statistical mechanism apparatus to processes
that are totally deterministic"
```

Corresponding keyphrases

```
"id": "36",
"tgt": ["game theory", "statistical mechanism apparatus",
        "incorrect banking data", "deterministic processes",
        "noncomplex numbers", "bagsik oscillator"]
```

Figure 6. Example source text and keyphrases

Under the condition of k = m, we apply CatSeq and Ranking-KPG to generate keyphrases for this article. Figure 7 shows the results of Ranking-KPG. It gives not only all the three present keyphrases predictions but also one accurate absent keyphrases prediction.

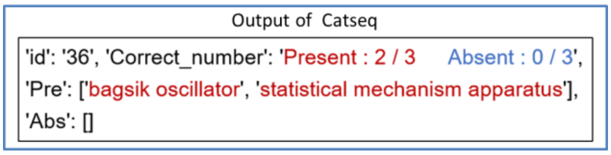


Output of Ranking-KPG

```
'id': '36', 'Correct_number': 'Present : 3 / 3    Absent : 1 / 3',
'Pre': ['bagsik oscillator', 'statistical mechanism apparatus', 'incorrect banking data']
'Abs': ['determinist process']
```

Figure 7. Output of Ranking-KPG

Figure 8 shows the generated results when CatSeq is used alone as the generator. CatSeq gives two accurate present keyphrase predictions for this article but does not provide any accurate absent keyphrases predictions.



Output of Catseq

```
'id': '36', 'Correct_number': 'Present : 2 / 3    Absent : 0 / 3',
'Pre': ['bagsik oscillator', 'statistical mechanism apparatus'],
'Abs': []
```

Figure 8. Output of Ranking-KPG

7. Conclusion and Future Work

For the keyphrase generation task, our proposed approaches based on finetuned BART improved the performance of absent keyphrase generation compare with all the baselines, in terms of F1-score over Inspec, Krapivin, NUS, and SemEval datasets.

For present keyphrase extraction, the combination model of BART and BK-Rank shows excellent performance on maximum recall rate, but the BERT cross-encoder-based ranking method still needs to be further improved.

For future work, we tend to continue to improve the ranking mechanism, improving the recall rate of absent keyphrases by utilizing masked language model and constructing domain corpus for scientific papers.

References

[1] Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, Martin Jaggi, "Simple Unsupervised Keyphrase Extraction using Sentence Embeddings", Proceedings of the 22nd Conference on Computational Natural Language Learning. Brussels, Belgium, pp.221–229, 2018.

[2] Hou Pong Chan, Wang Chen, Lu Wang, Irwin King. "Neural Keyphrase Generation via Reinforcement Learning with Adaptive Rewards". ACL 2019.

[3] S. Humeau, K. Shuster, M. A. Lachaux, et al. "Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring". International Conference on Learning Representations. 2019.

[4] Yang Liu, Mirella Lapata, "Text Summarization with Pretrained Encoders", EMNLP/IJCNLP. Hong Kong, China, pp. 3730–3740, 2019.

[5] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension", ACL, Online, pp. 7871–7880, 2020.

[6] Tingyi Liu, Mizuho Iwaihara, "Supervised Learning of Keyphrase Extraction Utilizing Prior Summarization", July 2021.

[7] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, Yu Chi, "Deep Keyphrase Generation", ACL, Vancouver, Canada, 2017, pp. 582–592.

[8] Mikalai Krapivin, Aliaksandra Autayeu, Maurizio Marchese, "Large dataset for keyphrases extraction", Technical Report DISI-09-055, DISI, Trento, Italy, 2008.

[9] Xingdi Yuan, Tong Wang, Rui Meng, "One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases", ACL. Online, pp. 7961–7975, 2018.