# Representation Learning for Resource-Constrained Keyphrase Generation

Anonymous ACL submission

## Abstract

State-of-the-art keyphrase generation methods generally depend on large annotated datasets, limiting their performance in domains with constrained resources. To overcome this challenge, we investigate pre-training strategies to learn an intermediate representation suitable for the keyphrase generation task. We introduce *salient span recovery* and *salient span prediction* as guided denoising language modeling objectives that condense the domain-specific knowledge essential for keyphrase generation. Through experiments on benchmarks spanning multiple domains, we show the effectiveness of the proposed approaches for facilitating low resource and zero-shot keyphrase generation.

## 1 Introduction

Keyphrases of a document are the phrases that identify and summarize the most important information. Given a document, the task of keyphrase generation requires the prediction of a set of keyphrases, each of which is classified as a *present keyphrase* if it appears in the document, or an *absent keyphrase* otherwise. The generated keyphrases can facilitate a wide range of applications, such as recommendation (Wu and Bolivar, 2008; Dave and Varma, 2010), text summarization (Zhang et al., 2004), text classification (Hulth and Megyesi, 2006; Wilson et al., 2005; Berend, 2011), document clustering (Hammouda et al., 2005), and information retrieval tasks (Jones and Staveley, 1999; Kim et al., 2013; Tang et al., 2017; Boudin et al., 2020).

Despite the promising results of keyphrase generation methods (Meng et al., 2017; Chen et al., 2018, 2019; Ahmad et al., 2021), they often require a large amount of annotated data. In real-world applications, the limited availability of such resources has proposed challenges beyond optimization and weak abilities to generalize. For instance, news topics emerge frequently and require precise recognition as keyphrases. The input genre and style may also change, leading to the domain shift effect.

In this paper, we focus on tackling the challenges of keyphrase generation in low resource settings by learning a domain-specific representation with unlabeled data. Motivated by the observation that keyphrases are often snippets or synonyms of the salient in-text information, and that such information can be identified by statistical methods, we design **salient span recovery (SSR)** and **salient span prediction (SSP)** to fine-tune BART (Lewis et al., 2020). Through corrupting the most salient parts of the input document, SSR and SSP encourage the model to focus on the information most important within domain and most conducive to the subsequent fine-tuning on the small dataset.

Through low-resource benchmarks covering various domains, we show the advantage of training in-domain intermediate representations[1]. Moreover, compared with other objectives such as title generation and text infilling, we find that salient span recovery achieves the best performance for both low resource absent keyphrase generation and zero-shot cross-domain transfer.

## 2 Methodology

**Problem Definition** We define a keyphrase generation dataset $D_{kp}$ as a set of tuples $(\mathbf{x^i}, \mathbf{p^i})$, where $\mathbf{x^i} = (x^i_1\ x^i_2\ ...\ x^i_{|\mathbf{x^i}|})$ is the $i$th input document, and $\mathbf{p^i} = \{p^i_1, p^i_2, ..., p^i_{|\mathbf{p^i}|}\}$ is the set of corresponding keyphrases. In addition, we introduce $D_{aux}$ to refer to the set of *unlabeled* documents from the same domain as $D_{kp}$. Following Yuan et al. (2020), we formulate keyphrase generation as generating a sequence of tokens that is the concatenation of the keyphrases $\mathbf{y^i} = (p^i_1$ `[sep]` $p^i_2$ `[sep]` ... `[sep]` $p^i_{|\mathbf{p^i}|})$[2] based on the source text $\mathbf{x^i}$.

[1] We will release the source code for reproducing our experiments upon paper acceptance.

[2] We use semicolon as `[sep]` in our implementation.

### 2.1 Intermediate Representation Learning

We argue that, in order to generate good keyphrases, both **intra-article** and **domain-wise** reasoning are necessary. Intra-article reasoning entails identifying, connecting, and abstracting spans in the article. In contrast, domain-wise reasoning determines whether a phrase is salient within a specific domain and thus qualifies as a keyphrase.

As $|D_{kp}|$ is small, directly fine-tuning the base BART model leads to sub-optimal performance. Therefore, we aim to leverage information in $D_{aux}$ to learn domain-specific intermediate representations before fine-tuning on $D_{kp}$.

One straightforward approach to use $D_{aux}$ is to continue performing **text infilling**, one of the objectives for pre-training BART (Lewis et al., 2020). However, it mainly focuses on intra-article reasoning, and may not efficiently model domain-wise knowledge. Alternatively, as suggested by Ye and Wang (2018), knowledge from **title generation** can benefit keyphrase generation. Indeed, title generation is a form of summarization which requires intra-document reasoning and to some extent uses domain-wise information. However, it fails to model the structure of keyphrases, hides the diversity of the keyphrase space, and is often of an extractive nature. Therefore, we propose the following task-specific pre-training loss for learning domain-specific intermediate representation.

**Salient Span Recovery** To condense the knowledge of both types of reasoning and to benefit absent keyphrase generation as much as present keyphrase generation, we design salient span recovery as a variant of text infilling objective where the tokens for masking are strategically chosen. Given $D_{aux}$, we first use TF-IDF to identify a set of n-grams $\{q_1^i, ..., q_n^i\}$ for each $\mathbf{x^i} \in D_{aux}$. During training, each occurrence of $q_j^i$ in $\mathbf{x^i}$ is replaced with a single `[MASK]` token with probability $k_s$. To create additional perturbation, we also mask each of words in $\mathbf{x^i} \setminus (q_1^i \cup ... \cup q_n^i)$ with probability $k_o$ to obtain the final input $\mathbf{x^i_{SSR}}$. The model is trained to minimize the cross entropy loss $\mathcal{L}_{CE}(\mathbf{z^i}, \mathbf{x^i})$, where $\mathbf{z^i}$ is the model's reconstruction of the corrupted input $\mathbf{x^i_{SSR}}$.

**Salient Span Prediction** To represent the structures of keyphrases more explicitly, we design SSP as an alternative to SSR. SSP's input is still $\mathbf{x^i_{SSR}}$, but the output is the concatenation of the TF-IDF predictions $\mathbf{x^i_{SSP}} = (q_1^i$ `[sep]` $q_2^i$ `[sep]` ... `[sep]` $q_n^i)$. The model is trained to minimize the cross entropy loss $\mathcal{L}_{CE}(\mathbf{z^i}, \mathbf{x^i_{SSP}})$, where $\mathbf{z^i}$ is the model's reconstruction of the corrupted input $\mathbf{x^i_{SSR}}$.

## 3 Experimental Setup

### 3.1 Datasets

We conducted experiments on two benchmarks. For each benchmark, we split the train set into a small $D_{kp}$ and a large $D_{aux}$, while keeping the validation and test sets the same. The statistics of test datasets we use are presented in the appendix.

**Scientific Articles**. We use KP20k (Meng et al., 2017) for training and evaluate on KP20k (test set), Inspec (Hulth, 2003a), Krapivin (Krapivin et al., 2009), NUS (Nguyen and Kan, 2007), and SemEval (Kim et al., 2010). After removing articles overlapping with the validation or test set, the KP20k train set contains 509,818 instances. We set $|D_{kp}| = 20,000$ for KP20k, i.e., only $20,000$ documents will be used for supervised training.

**News**. We use KPTimes (Gallina et al., 2019) for training and evaluation. After necessary preprocessing, the KPTimes train set contains 259,923 instances. We set $|D_{kp}| = 10,000$ for KPTimes.

### 3.2 Baseline and Evaluation Metrics

Using the $D_{kp}$, we fine-tune the pre-trained BART and its derivative models obtained by text infilling, title generation, salient span recovery, and salient span prediction. We also compare with a randomly initialized Transformer (Vaswani et al., 2017), TextRank (Hulth and Anette, 2004), and `AutoKeyGen` (Shen et al., 2021).

Following Chan et al. (2019), we use greedy decoding and report the F1@5 and F1@M for both present and absent keyphrases, where F1@$k$ only considers the top $k$ predictions, and F1@M takes all predictions from the model for evaluation. We do not report F1@M for TextRank and `AutoKeyGen` because the total number of predictions is a hyperparameter for these unsupervised methods. We repeat each experiment using three different seeds and report the average scores.

## 4 Results and Analysis

### 4.1 Intermediate Representation Learning

Table 1 and 2 show the performance of low resource absent keyphrase generation and present keyphrase generation with intermediate representation learning in the scientific domain.

| Method | KP20k | | Inspec | | Krapivin | | NUS | | Semeval | |
|---|---|---|---|---|---|---|---|---|---|---|
| | F1@5 | F1@M | F1@5 | F1@M | F1@5 | F1@M | F1@5 | F1@M | F1@5 | F1@M |
| Transformer | 0.96 | 1.47 | 0.31 | 0.46 | 1.16 | 1.76 | 1.02 | 1.38 | 0.95 | 1.18 |
| BART | 1.14 | 1.80 | 0.88 | 1.28 | 1.40 | 2.09 | 1.33 | 1.75 | 0.83 | 1.01 |
| BART+TI | 1.71 | 2.78 | 1.20 | 1.80 | 2.12 | 3.15 | 1.88 | 2.56 | 1.18 | 1.54 |
| BART+TG | 1.77 | 2.62 | 1.34 | 1.91 | 2.36 | 3.16 | 2.20 | 2.77 | 1.00 | 1.21 |
| BART+SSP | 1.89 | 3.11 | 1.14 | 1.63 | **2.87** | **4.31** | 2.30 | 2.93 | **1.46** | **1.83** |
| BART+SSR | **2.11** | **3.43** | **1.65** | **2.31** | 2.84 | 4.15 | **2.44** | **3.12** | 1.36 | 1.65 |

Table 1: F1 scores of absent keyphrase generation on five benchmarks from the scientific domain. "TI" = Text Infilling; "TG" = Title Generation; "SSP" = Salient Span Prediction; "SSR" = Salient Span Recovery. SSR and SSP outperform the other approaches in all benchmarks. Some example outputs are presented in the appendix.

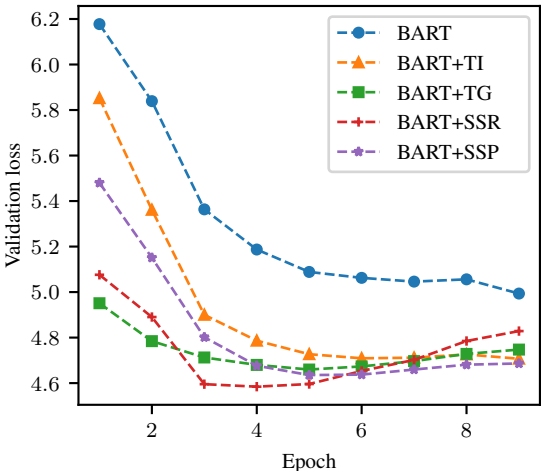


Figure 1: A comparison of KP20k low resource validation loss of BART with different initializions. BART+SSR converges to the lowest loss in 3 epochs.

**Baselines** From Table 1 and 2, it is apparent that fine-tuning BART significantly outperforms the Transformer trained from scratch, with the scores more than doubled for the four additional evaluation benchmarks. This shows the advantage of leveraging the pre-trained language model. On top of the pre-trained BART, performing domain-specific text infilling can further benefit both present and absent keyphrase generation on KP20k. By contrast, using the representation learned with title generation achieves the best low resource present keyphrase generation performance. However, its absent keyphrase generation performance is worse than most of the other methods (except for Inspec). Intuitively, titles summarize and emphasize the most salient message of the articles, and tend to be extractive instead of abstractive.

**Salient Span Recovery** According to Table 1 and 2, SSR is effective for improving both present and absent keyphrase generation performance compared to text infilling, achieving the highest absent keyphrase performance and the second highest present keyphrase performance on KP20k and most of the four evaluation benchmarks. In addition, we find predictions of BART+SSR generally having higher relevance to the input. We include some of the qualitative results in the appendix.

Figure 1 presents the validation loss for low resource fine-tuning. We observe that all intermediate representation learning methods we study outperform the BART fine-tuning baseline. Initializing with salient span recovery converges the fastest and achieves the best validation loss. In addition, we find that salient span recovery consistently outperforms salient span prediction. One reason may be that the quality of the keyphrases obtained using TF-IDF may be too low to be used as-is like manually annotated keyphrase labels. We provide additional results on KPTimes in the appendix.

### 4.2 Zero-Shot Adaptation Performance

After confirming the effectiveness of the intermediate representations on facilitating low resource training, we continue to experiment with zero-shot adaptation. With $D_{kp}$ replaced by the KPTimes train set and $D_{aux}$ still being the KP20k train set, we then measure and compare the performance of the methods on the KP20k test set.

The results are presented in Table 3. Although no manual labels are used in the intermediate training, the learned representation indeed condenses domain-specific knowledge, which results in better zero-shot transfer performance. SSR achieves the best zero-shot performance, outperforming the other methods by a large margin in all metrics. We also report the score of predictions from the intermediate SSP model. Despite competitive performance on present keyphrases, its absent keyphrase performance is worse than the baseline.

### 4.3 Can TF-IDF Better Indicate Saliency?

Although we have shown the effectiveness of intermediate representations trained with SSR and SSP, it is still worth understanding whether TF-IDF actually captures domain-wise saliency knowledge. Therefore, we compute the overlap between

| Method | KP20k | | Inspec | | Krapivin | | NUS | | Semeval | |
|---|---|---|---|---|---|---|---|---|---|---|
| | F1@5 | F1@M | F1@5 | F1@M | F1@5 | F1@M | F1@5 | F1@M | F1@5 | F1@M |
| TextRank | 18.1 | N/A | 26.3 | N/A | 14.8 | N/A | 18.7 | N/A | 16.8 | N/A |
| AutoKeyGen | 23.4 | N/A | **30.3** | N/A | 17.1 | N/A | 21.8 | N/A | 18.7 | N/A |
| Transformer | 8.60 | 13.31 | 3.35 | 4.30 | 6.72 | 10.50 | 9.99 | 13.78 | 7.12 | 9.51 |
| BART | 21.36 | 25.61 | 22.10 | 25.58 | 20.45 | 22.68 | 26.28 | 28.91 | 21.89 | 23.54 |
| BART+TI | 24.23 | 28.80 | 21.18 | 24.20 | 21.18 | 22.27 | 28.12 | 29.45 | 21.59 | 23.01 |
| BART+TG | **27.97** | **31.29** | 25.01 | **28.93** | **25.28** | **27.86** | **32.68** | **35.35** | **26.00** | **28.25** |
| BART+SSP | 25.02 | 28.51 | 21.62 | 24.60 | 22.13 | 22.99 | 29.44 | 31.19 | 25.01 | 27.24 |
| BART+SSR | 26.32 | 29.76 | 22.24 | 25.29 | 24.39 | 25.20 | 31.02 | 32.64 | 23.47 | 24.84 |

Table 2: F1 scores of present keyphrase generation on five benchmarks from the scientific domain. BART+TG achieves the best performance on most benchmarks, while BART+SSR also gives competitive scores. We use the scores of TextRank and AutoKeyGen reported by Shen et al. (2021).

| Method | Present | | Absent | |
|---|---|---|---|---|
| | F1@5 | F1@M | F1@5 | F1@M |
| BART | 3.41 | 5.28 | 0.16 | 0.19 |
| SSP-only | 8.76 | 8.96 | 0.13 | 0.17 |
| BART+TI | 7.21 | 11.05 | 0.26 | 0.34 |
| BART+TG | 5.91 | 9.02 | 0.26 | 0.31 |
| BART+SSP | 7.09 | 10.82 | 0.32 | 0.41 |
| BART+SSR | **9.75** | **14.28** | **0.40** | **0.56** |

Table 3: F1 scores of zero-shot keyphrase generation on KP20k. "SSP-only" = the SSP model on KP20k. BART+SSR significantly outperforms other methods.

TF-IDF's prediction (or titles) and the manually annotated keyphrases as a proxy measure. We define **phrase recall** as the proportion of present keyphrases that are also identified by TF-IDF or titles, **word recall** as the proportion of all words in present keyphrases that are also identified by TF-IDF or titles, and **word precision** as the proportion of words in TF-IDF's predictions or titles that are included in any keyphrase of the same document.

As presented in Table 4, compared to document titles, TF-IDF's predictions have high phrase recall and word recall with lower word precision. Salient span recovery fully takes advantage of this high coverage to exercise a wide range of keyphrase-related salient information. Meanwhile, the false positives of TF-IDF are converted into non-harmful random masks during training.

## 5 Related Work

**Low Resource Keyphrase Generation** Automatic keyphrase generation has been a popular topic of study. While keyphrase extraction only extracts present keyphrases as spans of the document (Hulth, 2003b; Mihalcea and Tarau, 2004; Wan and Xiao, 2008; Zhang et al., 2016), keyphrase generation directly predicts both types of keyphrases (Meng et al., 2017; Chen et al., 2018, 2019; Zhao and Zhang, 2019; Chan et al., 2019; Yuan et al., 2020; Swaminathan et al., 2020; Ahmad et al., 2021; Ye et al., 2021). However, there are only a few works on low resource keyphrase generation. Ye and Wang (2018) used synthetic labeling and multitask learning to leverage large unlabeled datasets. Lancioni et al. (2020) used reinforcement learning to exploit learning signals from a pre-trained discriminator in the setting of Generative Adversarial Networks.

| Metric | KP20k | | KPTimes | |
|---|---|---|---|---|
| | Title | TF-IDF | Title | TF-IDF |
| **Phrase Recall** | 0.2553 | **0.4184** | 0.1223 | **0.2673** |
| **Word Recall** | 0.5441 | **0.8064** | 0.2829 | **0.6355** |
| **Word Precision** | **0.3937** | 0.1730 | **0.2929** | 0.1164 |

Table 4: An analysis of overlaps with present keyphrases for titles and TF-IDF predictions.

**Language Modeling for Low Resource Learning** Recent studies have successfully used pre-trained language models for rich-resource keyphrase generation (Liu et al., 2021) and keyphrase extraction (Sahrawat et al., 2019). Meanwhile, for various other tasks, studies explored continued domain-adaptive pre-training of the autoencoding (Gururangan et al., 2020; Lee et al., 2019) and encoder-decoder language models (Yu et al., 2021). Our approach belongs to the latter type. Our masking granularity is most similar to Lewis et al. (2020) and Joshi et al. (2020), while our span selection is most similar to Guu et al. (2020). Different from Guu et al. (2020), our approach infers *domain-adaptive* masks and mainly uses *phrase-level* infilling.

## 6 Conclusion

This paper considers the problem of low resource keyphrase generation. We show that learning an in-domain intermediate representation greatly facilitates fine-tuning with constrained resources. We design salient span recovery and salient span prediction as intermediate objectives and verify their effectiveness in both low resource and zero-shot scenarios. Future works may consider extending this work by composing the intermediate objectives and combining the representation learning techniques with a data-oriented approach.

## 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.

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

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

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, Yifan Gao, Jiani Zhang, Irwin King, and Michael R. Lyu. 2019. Title-guided encoding for keyphrase generation. In *AAAI*.

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

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.

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

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training.

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

Hulth and Anette. 2004. Textrank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP '04, pages 404–411. Association for Computational Linguistics.

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

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

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

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

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

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

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

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

Giuseppe Lancioni, Saida S.Mohamed, Beatrice Portelli, Giuseppe Serra, and Carlo Tasso. 2020. Keyphrase generation with GANs in low-resources scenarios. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 89–96, Online. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *CoRR*, abs/1901.08746.

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

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

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

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

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

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

Xianjie Shen, Yinghan Wang, Rui Meng, and Jingbo Shang. 2021. Unsupervised deep keyphrase generation. *ArXiv*, abs/2104.08729.

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

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

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

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

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

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

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

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

Tiezheng Yu, Zihan Liu, and Pascale Fung. 2021. AdaptSum: Towards low-resource domain adaptation for abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5892–5904, 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.

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

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

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

# Supplementary Material: Appendices

## A Test Set Statistics

| Dataset | #Examples | #KP | \|KP\| | %AKP |
|---|---|---|---|---|
| KP20k | 20000 | 5.28 | 2.04 | 37.06 |
| Inspec | 500 | 9.83 | 2.48 | 26.38 |
| Krapivin | 400 | 5.85 | 2.21 | 44.34 |
| NUS | 211 | 11.65 | 2.22 | 45.61 |
| SemEval | 100 | 14.66 | 2.38 | 57.37 |
| KPTimes | 20000 | 5.03 | 2.00 | 37.84 |

Table 5: Statistics of all the test sets we use. #KP: average number of keyphrases of each instance; |KP|: average length of each keyphrase; %AKP: the percentage of absent keyphrases.

## B Implementation Details

We use Fairseq's[3] BART-base implementation and its pre-trained checkpoint to conduct the experiments. BART-base has about 140 million parameters in total with 6 encoder and 6 decoder layers and hidden size 768. We truncate the input documents to 1024 tokens. We use Adam optimizer with momentum with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and polynomial decay with 1000 warm-up steps. The initial learning rate is set to 0.00003, and we use effective batch size of 64. For each experiment, we use the validation dataset of KP20k and KPTimes to choose the best checkpoint. We use greedy decoding to generate predictions until the EOS token is generated. To encourage the model to generate more keyphrases, we prohibit the generation of the EOS token until 16 tokens have been generated. We use the same optimizations parameters for both the intermediate representation learning and fine-tuning on keyphrase generation. All experiments are run on two GTX 1080Ti GPUs.

We implement the objectives for intermediate representation learning in the following manner.

**Salient Span Recovery and Salient Span Prediction.** We adapt the implementation in this repository to obtain TF-IDF predictions. We gather phrases up to trigrams and generate 30 n-grams per document. During training, we use $k_s = 0.8$ and $k_o = 0.2$. We run the mask generation algorithm offline to prepare data for each epoch, and use Fairseq's `translation` task and the sharding functionality for training.

**Text Infilling**. Given $\mathbf{x^i}$, text infilling randomly selects spans with lengths following a Poisson distribution ($\lambda = 3$), and replaces the span with a single `[MASK]` token to obtain $\mathbf{x^i_{Infilling}}$. The model is trained to minimize the cross entropy loss $\mathcal{L}_{CE}(\mathbf{z^i}, \mathbf{x^i})$, where $\mathbf{z^i}$ is the model's reconstruction of the corrupted input $\mathbf{x^i_{Infilling}}$. We use Fairseq's `denoising` task for training.

**Title Generation**. We remove the titles from $\mathbf{x^i}$ and fine-tune BART for generating the titles. The model is trained to minimize the cross entropy loss between the titles and the model's prediction based on the articles without titles. We use Fairseq's `translation` task for training.

[3] https://github.com/pytorch/fairseq

## C Performance versus Resource

To define the resource-constrained scenarios and to find whether learning is equally sensitive to increasing resource in all resource settings, we generate subsets of KP20k with different sizes, and directly fine-tune BART on them. Figure 2 shows how test performance improves as more training data is given. We observe that for both present and absent keyphrase generation, the performance increases sharply before gradually levels off. In addition, for KP20k, we find the growth rate of present keyphrase generation performance scales better with resource, while the growth of absent keyphrase generation slows down after 200,000 documents. Therefore, for each scenario, we pick roughly 4% of the entire dataset as the size of the low resource training set. As a side remark, with the full train set of KP20k, we obtain 0.37 F1@M for present keyphrases and 0.04 F1@M for absent keyphrases, which is on par with previous works such as Yuan et al. (2020); Swaminathan et al. (2020); Ye et al. (2021).

## D Results on KPTimes

We report the results on KPTimes in Table 6. Different from the scientific domain, we found that title generation results in the best downstream fine-tune performance for both present and absent keyphrases. This suggests that the help from title generation can be domain-dependent.

## E Example Outputs

We present two set of outputs in Figure 3 and Figure 4. Figure 3 presents the predictions of zero-shot models on KP20k (corresponding to Table 3). Figures 4 presents the predictions of the low resource

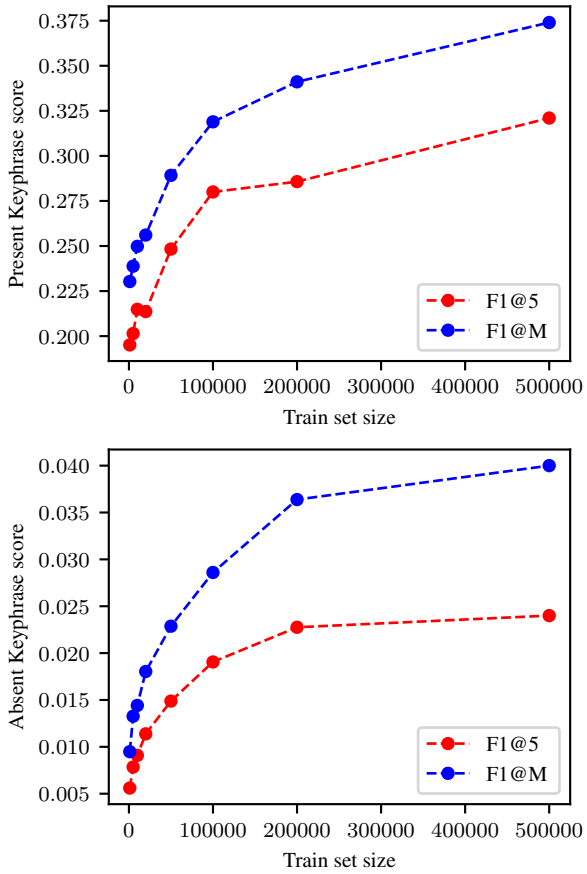


Figure 2: KP20k test performance of BART fine-tuned on subsets of KP20k with difference sizes. For each size, we repeat the experiment for three times.

models on the scentific benchmark datasets (corresponding to Table 1 and 2).

## F Limitations and Ethical Statement

One core assumption of this work is that BART is a competitive pre-trained model for keyphrase generation and has its unique advantages for domain-adaptive representation learning. We consider BART instead of other popular models such as T5 because T5 uses fill-in-the-blank task while BART uses denoising-autoencoding, which is by nature closer to salient span recovery and salient span prediction.

We note that our approach involves large-scale unlabeled data, which may introduce additional bias. As our approach can be easily integrated into BART-based keyphrase generation services, we encourage the potential users to monitor for the potential biases closely and apply corresponding bias-mitigation measures when necessary.

**Computational Budget** All experiments are run on a local GPU server. On average, the intermediate pre-training stage takes 20 to 30 GPU hours on a dataset of size similar to KP20k, and the final tine-tuning stage takes less than 1 GPU hour on a dataset with less than 20,000 examples. We acknowledge that the large-scale representation learning may lead to additional energy cost and emissions. However, our approach justifies the cost by (1) having better performance in solving the challenge low resource problem and (2) allowing the resulting domain-specific representation to be reused for fine-tuning on different low resource datasets.

| Method | Present | | Absent | |
|---|---|---|---|---|
| | F1@5 | F1@M | F1@5 | F1@M |
| Transformer | 15.73 | 24.55 | 8.04 | 10.86 |
| BART | 23.39 | 33.15 | 11.72 | 15.22 |
| BART+TI | 25.26 | 35.66 | 13.05 | 16.92 |
| BART+TG | **29.31** | **41.00** | **14.16** | **18.79** |
| BART+SSP | 20.13 | 30.34 | 10.20 | 11.93 |
| BART+SSR | 24.13 | 34.11 | 12.83 | 16.75 |

Table 6: Keyphrase generation performance on KPTimes. The models are fine-tuned on an 10k document low resource subset. We repeat each experiment using three different splits and report the average scores.

**Artifact and Licensing** The KP20k dataset and the Fairseq library we use are MIT licensed, and the KPTimes dataset is Apache 2.0 licensed. While commercial use is allowed for these artifacts, we only use them for research. We will make our code and models publicly available after the anonymity period. In addition, we will not re-distribute the datasets. Instead, we will refer to their original hosts.

**Data Anonymizing** We use the KP20k and KPTimes datasets distributed by their original hosts. We did not systematically examine for the sensitive information because similar inspections have been done by previous work and by the original authors of the dataset. We have verified that our pre-processing methods do not introduce external biases or sensitive information.

**Title:** polynomial algorithms for partitioning problems on graphs with fixed clique width ( extended abstract ) .
**Abstract:** we consider three graph partitioning problems , both from the vertices and the edges point of view . these problems are dominating set , list q coloring with costs ( fixed number of colors q ) and coloring with non fixed number of colors . they are all known to be np hard in general . we show that all these problems ( except edge coloring ) can be solved in polynomial time on graphs with clique width bounded by some constant k , if the k expression of the input graph is also given . in particular , we present the first polynomial algorithms ( on these classes ) for chromatic number , edge dominating set and list q coloring with costs ( fixed number of colors q , both vertex and edge versions ) . since these classes of graphs include classes like p [digit] sparse graphs , distance hereditary graphs and graphs with bounded treewidth , our algorithms also apply to these graphs .
**Ground Truth:** edge coloring ; dominating set ; clique width ; polynomial algorithms ; coloring ; edge dominating set
**BART+TG:** polynomialism ; tech industry ; computers and the internet ; computer and video games
**BART+TI:** tech industry ; polynomials ; graph ; computer security ; computers and the internet
**BART+SSP:** polynomial time ; vertex ; clique ; coloring ; nyc ; trees
**BART+SSR:** dominating set ; clique width ; polynomial algorithms ; list q coloring ; edge dominating set ; vertex

**Title:** extending record typing to type parametric modules with sharing .
**Abstract:** we extend term unification techniques used to type extensible records in order to solve the two main typing problems for modules in standard ml matching and sharing . we obtain a type system for modules based only on well known unification problems , modulo some equational theories we define . our formalization is simple and has the elegance of polymorphic type disciplines based on unification . it can be seen as a synthesis of previous work on module and record typing .
**Ground Truth:** ml ; parametric ; module ; extensibility ; matching ; order ; type system ; sharing ; synthesis ; records ; unification ; theory ; standardization ; formalism ; polymorphic
**BART+TG:** data storage ; computer security ; computers and the internet ; typing ( sports )
**BART+TI:** lambda lambi ; curry gilbert ; curry curry ; curry howard ; curry ; curry raster ; lambda phillips ; curry jack
**BART+SSP:** language and languages ; instant replay ( sports ; software ; computer security ; instant messaging ; unification
**BART+SSR:** ml ; module ; unification ; haskell ; type system ; inheritance and estate tax

**Title:** localization and regularization behavior of mixed finite elements for 2d structural problems with damaging material .
**Abstract:** a class of lagrangian mixed finite elements is presented for applications to 2d structural problems based on a damage constitutive model . attention is focused on localization and regularization issues as compared with the correspondent behavior of lagrangian displacement based elements . a non local regularization procedure of integral type is adopted . a predictorcorrector technique is used to solve the evolution problem of the damage variable . the proposed elements show superior performances for typical structural applications .
**Ground Truth:** localization ; hybrid formulations ; mixed finite elements ; damage ; regularization ; plasticity
**BART+TG:** science and technology ; nikkei technology ; engineering ; engineering and engineers
**BART+TI:** science and technology ; federal element ; lagrangian - displacement - brick and tile ; engineering ; buildings
**BART+SSP:** localization ; science and technology ; lagrangian element ; engineering ; engineering and engineers ; element
**BART+SSR:** localization ; elastoplasticity ; engineering ; mixed finite elements ; damage

**Title:** nature inspired techniques for conformance testing of object oriented software .
**Abstract:** soft computing offers a plethora of techniques for dealing with hard optimization problems . in particular , nature based techniques have been shown to be very efficient in optimization applications . the present paper investigates the suitability of various nature inspired meta heuristics ( genetic algorithms , evolutionary programming and ant colony systems ) to the problem of software testing . the present study is part of the nature inspired techniques for object oriented testing ( nitot ) environment . it aims at addressing the problem of conformance testing of object oriented software to its specification expressed in terms of finite state machines . detailed description , adaptation and evaluation of the various nature inspired meta heuristics are discussed showing their potential in this context of conformance testing .
**Ground Truth:** evolutionary programming ; ant colony systems ; genetic algorithms ; testing data generation ; conformance testing
**BART+TG:** tech industry ; software ; nature inspired techniques for object oriented testing ; computers and the internet
**BART+TI:** science and technology ; nature inspired techniques for object oriented testing ( nitot ; software ; computers and the internet ; tests and testing
**BART+SSP:** nature ; tests and testing ; object oriented ( theory and philosophy ; software
**BART+SSR:** evolutionary programming ; nature ; ant colony systems ; con protocol ; genetic algorithms ; software ; object oriented software

**Title:** compressible distributions for high dimensional statistics .
**Abstract:** we develop a principled way of identifying probability distributions whose independent and identically distributed realizations are compressible , i.e. , can be well approximated as sparse . we focus on gaussian compressed sensing , an example of underdetermined linear regression , where compressibility is known to ensure the success of estimators exploiting sparse regularization . we prove that many distributions revolving around maximum a posteriori ( map ) interpretation of sparse regularized estimators are in fact incompressible , in the limit of large problem sizes . we especially highlight the laplace distribution and regularized estimators such as the lasso and basis pursuit denoising . we rigorously disprove the myth that the success of minimization for compressed sensing image reconstruction is a simple corollary of a laplace model of images combined with bayesian map estimation , and show that in fact quite the reverse is true .
**Ground Truth:** linear inverse problems ; statistical regression ; maximum a posteriori estimator ; order statistics ; lasso ; basis pursuit ; instance optimality ; sparsity ; compressible distribution ; compressed sensing ; high dimensional statistics
**BART+TG:** lasso lasso ; basis pursuit denos ; pursuit denoising ; gaussian compressed sensing ; statistics
**BART+TI:** lasso ; sparsity ; spurs ; statistics ; spanish language
**BART+SSP:** space ; gaussian ; bayesian photography ; distribution ; john j p chase & co ; photography ; lasso john c
**BART+SSR:** denocings ; lasso ; basis pursuit ; sparsity ; data mining , big data ; gaussian compressed sensing ; statistics ; compressible distribution ; denoising

Figure 3: Example zero-shot transfer outputs on the scentific benchmarks. Correct keyphrases are colored in blue. "TI" = Text Infilling; "TG" = Title Generation; "SSP" = Salient Span Prediction; "SSR" = Salient Span Recovery.

**Title:** bounded skew clock and steiner routing under elmore delay .
**Abstract:** we study the minimum cost bounded skew routing tree problem under the elmore delay model . we present two approaches to construct bounded skew routing trees ( i ) the boundary merging and embedding ( bme ) method which utilizes merging points that are restricted to the boundaries of merging regions , and ( ii ) the interior merging and embedding ( ime ) algorithm which employs a sampling strategy and dynamic programming to consider merging points that are interior to , rather than on the boundary of , the merging regions . our new algorithms allow accurate control of elmore delay skew , and show the utility of merging points inside merging regions .
**Ground Truth:** pathlength delay ; bounded skew ; elmore delay ; vlsi ; routing trees ; global routing ; zero skew ; clock routing
**BART+TG:** bounded skew clock ; boundary merging and embedding ; boundary algorithms ; elmore delay ; boundary matching ; steiner routing
**BART+TI:** dynamic programming ; boundary merging ; sampling ; routing ; model ; routing tree ; embedding ; control ; clock skew ; region ; tree ; clock ; skew ; rier delay
**BART+SSP:** dynamic programming ; clock skew ; steiner tree ; bounded skew ; elmore delay ; clock
**BART+SSR:** dynamic programming ; boundary merging and embedding ; accuracy ; routing tree ; control ; bounded skew ; elmore delay ; dynamic program ; ampling strategy ; clock routing

**Title:** rsa oaep is secure under the rsa assumption .
**Abstract:** recently victor shoup noted that there is a gap in the widely believed security result of oaep against adaptive chosen ciphertext attacks . moreover , he showed that , presumably , oaep can not be proven secure from the one wayness of the underlying trapdoor permutation . this paper establishes another result on the security of oaep . it proves that oaep offers semantic security against adaptive chosen ciphertext attacks , in the random oracle model , under the partial domain one wayness of the underlying permutation . therefore , this uses a formally stronger assumption . nevertheless , since partial domain one wayness of the rsa function is equivalent to its ( full domain ) <unk> , it follows that the security of rsa oaep can actually be proven under the sole rsa assumption , although the reduction is not tight .
**Ground Truth:** oaep ; public key encryption ; rsa ; provable security
**BART+TG:** random oracle model ; one wayness ; rsa function ; onewayness ; chosen ciphertext attacks ; **rsas** ; security
**BART+TI:** random oracle model ; chosen ciphertext attack ; sa assumption ; partial domain one wayness ; one way security
**BART+SSP:** random oracle model ; semantic security ; chosen ciphertext ; rsa ; partial domain one wayness
**BART+SSR:** random oracle model ; one wayness ; chosen ciphertext attacks ; rsa ; provable security

**Title:** modularity in technology and organization .
**Abstract:** literature on property rights to create the outlines of a modularity theory of the firm . such a theory will look at firms , and other organizations , in terms of the partitioning of rights understood as protected spheres of authority among cooperating parties . it will assert that organizations reflect nonmodular structures , that is , structures in which decision rights , rights of alienation , and residual claims to income do not all reside in the same hands
**Ground Truth:** organization ; property rights ; nonmodular structures ; technology ; authority ; cooperating parties ; transaction costs ; partitioning of rights ; decision rights ; rights of alienation ; modularity
**BART+TG:** property property rights and property rights ; property rights ; property ; modularity
**BART+TI:** property rights ; income ; property property rights ; alienation ; partitioning ; alienation and residual claims ; modularity
**BART+SSP:** intellectual capital ; intellectual ownership ; spheres ; intellectual structure ; intellectual hierarchy ; intellectual property ; modularity ; intellectual organization ; technology ; protection ; intellectual assets ; intellectual asset ; intellectual equity ; intellectual properties ; organization ; property rights ; organizational structures ; intellectual rights ; intellectual space
**BART+SSR:** organization ; property rights ; partition ; political ; hierarchic ; ownership ; technology ; authority ; informal ; structure ; property ; firm ; property property ; modularity

**Title:** modelling user acceptance of building management systems .
**Abstract:** a questionnaire survey . these systems are crucial for optimising building performance and yet it has been widely reported that users are not making full use of their systems ' facilities . established models of technology acceptance have been employed in this research , and the positive influence of user perceptions of ease of use and compatibility has been demonstrated . previous research has indicated differing levels of importance of perceived ease of use relative to other factors . here , perceived ease of use is shown generally to be more important , though the balance between this and compatibility is moderated by the user perceptions of voluntariness
**Ground Truth:** information systems ; questionnaire survey ; compatibility ; innovation characteristics ; technology acceptance model ; voluntariness ; user perceptions ; ease of use ; user acceptance modelling ; building management systems
**BART+TG:** user acceptance ; building management systems ; modelling ; building performance ; building behaviour
**BART+TI:** technology compatibility ; voluntariness ; technology use ; technology acceptance ; building management systems ; building performance
**BART+SSP:** user perceptions of ease of use ; compatibility ; technology acceptance ; building management systems ; modelling
**BART+SSR:** questionnaire survey ; compatibility ; technology acceptance model ; usability ; building management systems ; modelling

Figure 4: Example outputs from low resource models on the scentific benchmarks. Correct keyphrases are colored in blue. "TI" = Text Infilling; "TG" = Title Generation; "SSP" = Salient Span Prediction; "SSR" = Salient Span Recovery.