A Joint Learning Approach based on Self-Distillation for Keyphrase Extraction from Scientific Documents

Tuan Manh Lai [1], Trung Bui [2], Doo Soon Kim [2], Quan Hung Tran [2]
[1] University of Illinois at Urbana-Champaign
[2] Adobe Research, San Jose, CA

Abstract

Keyphrase extraction is the task of extracting a small set of phrases that best describe a document. Most existing benchmark datasets for the task typically have limited numbers of annotated documents, making it challenging to train increasingly complex neural networks. In contrast, digital libraries store millions of scientific articles online, covering a wide range of topics. While a significant portion of these articles contain keyphrases provided by their authors, most other articles lack such kind of annotations. Therefore, to effectively utilize these large amounts of unlabeled articles, we propose a simple and efficient joint learning approach based on the idea of self-distillation. Experimental results show that our approach consistently improves the performance of baseline models for keyphrase extraction. Furthermore, our best models outperform previous methods for the task, achieving new state-of-the-art results on two public benchmarks: Inspec and SemEval-2017.

1 Introduction

Keyphrase extraction is the task of automatically extracting a set of representative phrases from a document that concisely describe its content. As keyphrases provide a brief yet precise description of a document, they can be utilized for various downstream applications (D'Avanzo and Magnini, 2005; Litvak and Last, 2008; Kim et al., 2010; Boudin and Morin, 2013). Over the past years, researchers have proposed many methods for the task, which can be divided into two major categories: supervised (Sterckx et al., 2016; Zhang et al., 2017; Alzaidy et al., 2019) and unsupervised techniques (Florescu and Caragea, 2017b; Boudin, 2018; Mahata et al., 2018). In the presence of sufficient domain-specific labeled data, supervised keyphrase extraction methods are often reported to outperform unsupervised methods (Kim et al., 2013; Caragea et al., 2014; Sahrawat et al., 2020).

Recently, many deep learning based methods have achieved promising performance in a wide range of NLP tasks (Lai et al., 2018b; Tran et al., 2018; Peters et al., 2018; Devlin et al., 2019; Lewis et al., 2019; Lai et al., 2020). However, most existing benchmark datasets for keyphrase extraction typically have limited numbers of annotated documents, making it challenging to train an effective deep learning model for the task. In contrast, digital libraries store millions of scientific articles online, covering a wide range of topics. While a significant portion of these articles have author-provided keyphrases, most other articles lack such kind of annotations. For example, major NLP conferences (i.e., ACL, EMNLP, COLING) normally do not require authors to provide keywords of their publications. In this paper, to effectively utilize these large amounts of unlabeled articles available online, we propose a novel joint learning approach based on the idea of self-distillation (Furlanello et al., 2018). To evaluate the effectiveness of our method, we use the Inspec and SemEval-2017 datasets. Experimental results show that our approach consistently improves the performance of baseline models. Furthermore, our best models achieve new state-of-the-art results on the two public benchmarks.

Proceedings of the 28th International Conference on Computational Linguistics, pages 649–656
Barcelona, Spain (Online), December 8-13, 2020

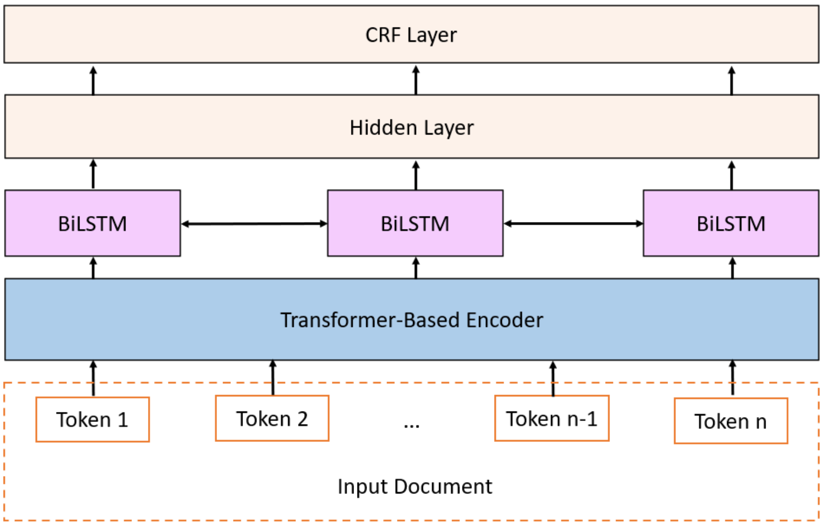


Figure 1: A high-level overview of our baseline model.

In summary, we have made the following contributions: (1) We propose a novel joint learning framework based on self-distillation for improving keyphrase extraction systems (2) Experiments on two public datasets, Inspec and SemEval-2017, demonstrate the effectiveness of our proposed methods. In the following parts, we first describe some preliminaries relating to the formulation of the keyphrase extraction problem and the architecture of our baseline models (Section 2). We then go into details at our self-distillation approach in Section 3. After that, we describe the conducted experiments and their results in Section 4. Finally, we conclude this work in Section 5.

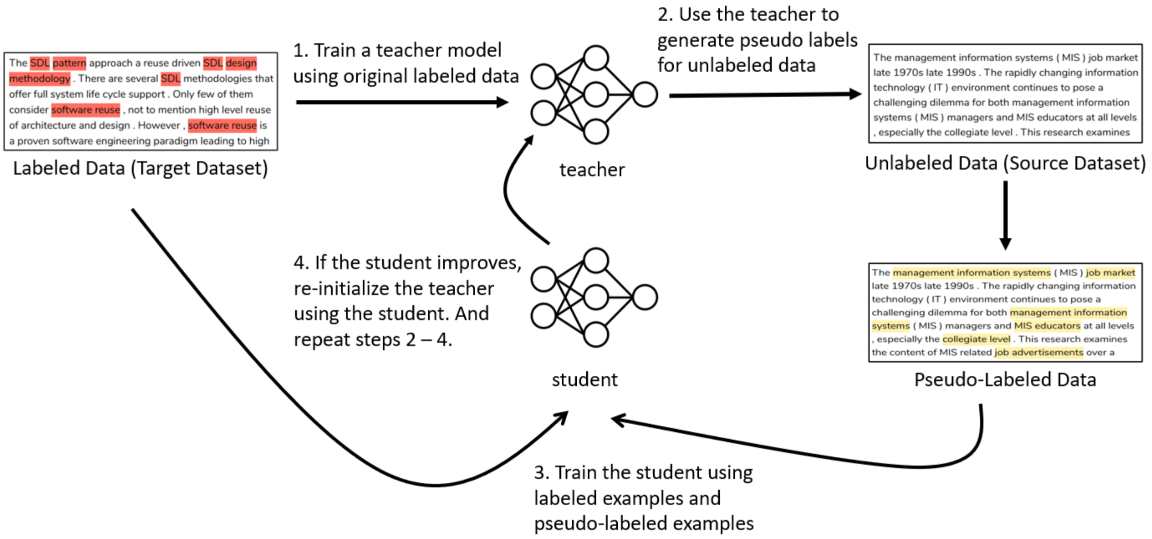
2 Preliminaries

Problem Formulation Similar to recent works (Gollapalli et al., 2017; Sahrawat et al., 2020), we formulate keyphrase extraction as a sequence labeling task. Let $D = (t_1, t_2, ..., t_n)$ be a document consisting of n tokens, where t_i represents the i^{th} token of the document. The task is to predict a sequence of labels $\mathbf{y} = (y_1, y_2, ..., y_n)$, where $y_i \in \{\texttt{I}, \texttt{B}, \texttt{O}\}$ is the label corresponding to token t_i. Label `B` denotes the beginning of a keyphrase, `I` denotes the continuation of a keyphrase, and `O` corresponds to tokens that are not part of any keyphrase. An advantage of this formulation is that it completely avoids the candidate generation step required in previous ranking-based approaches (El-Beltagy and Rafea, 2009; Bennani-Smires et al., 2018). Instead of having to generate a list of candidate phrases and then ranking them, we directly predict the target outputs in one go. This formulation also provides a unified approach to keyphrase extraction, as it has the same format as other sequence labeling tasks.

Baseline Models In this work, we employ the BiLSTM-CRF architecture as the baseline architecture (Huang et al., 2015; Alzaidy et al., 2019; Sahrawat et al., 2020; Zhu et al., 2020). Figure 1 shows a high-level overview of our baseline model. Given a sequence of input tokens, the model first forms a contextualized representation for each token using a Transformer-based encoder. The model then further uses a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) on top of the Transformer-based representations. After that, a dense layer is used to map the output of the bidirectional LSTM to the label space. Finally, a linear-chain CRF is applied to decode the labels.

3 Joint Learning based on Self-Distillation (JLSD)

Figure 2 shows a high-level overview of our proposed self-distillation approach. We refer to the labeled dataset as the target dataset and the unlabeled dataset as the source dataset. We first train a teacher model using existing labeled examples. After that, we start training a student model parameterized identically to the teacher model. During each training iteration, we sample a batch of both original



The SDL pattern approach a reuse driven SDL design methodology . There are several SDL methodologies that offer full system life cycle support . Only few of them consider software reuse , not to mention high level reuse of architecture and design . However , software reuse is a proven software engineering paradigm leading to high

The management information systems (MIS) job market late 1970s late 1990s . The rapidly changing information technology (IT) environment continues to pose a challenging dilemma for both management information systems (MIS) managers and MIS educators at all levels , especially the collegiate level . This research examines

The management information systems (MIS) job market late 1970s late 1990s . The rapidly changing information technology (IT) environment continues to pose a challenging dilemma for both management information systems (MIS) managers and MIS educators at all levels , especially the collegiate level . This research examines the content of MIS related job advertisements over a

Labeled Data (Target Dataset)
1. Train a teacher model using original labeled data
teacher
2. Use the teacher to generate pseudo labels for unlabeled data
Unlabeled Data (Source Dataset)
4. If the student improves, re-initialize the teacher using the student. And repeat steps 2 – 4.
student
Pseudo-Labeled Data
3. Train the student using labeled examples and pseudo-labeled examples

Figure 2: A high-level overview of our proposed method.

labeled examples and unlabeled examples with pseudo-labels generated by the teacher. At any point during the training process, if the student model's performance improves (i.e., achieves better results on the validation set of the target dataset), we re-initialize the teacher model with the current parameters of the student model, and then continue training the student using the same procedure. The intuition is to produce a virtuous cycle in which both the teacher and the student will become better together. A better teacher will generate more accurate pseudo-labels, which in turn helps train a better student. And then the improved student will be used to re-initialize the teacher. Algorithm 1 formally describes our proposed approach. Note that T denotes the number of training iterations, and r is a hyperparameter that determines how many unlabeled documents to be sampled in each iteration. During an iteration of JLSD, if the teacher happens to generate bad pseudo-labels for an example, the effect will not be detrimental. By design, the performance of the teacher is monotonically non-decreasing with time, therefore, when the teacher comes across the same example again, it is like to generate better pseudo-labels. To the best of our knowledge, the only other work exploring self-learning for keyphrase extraction is that of Zhu et al. (2020). However, during each training epoch, their method needs to label all unlabeled examples and then selects the ones with high confidence scores. This would incur a lot of overhead for every single training epoch as the number of unlabeled examples is typically very large. Our approach does not suffer from these issues. Our teacher model generates pseudo labels on-the-fly during each training iteration. Another highly related work is the paper by Ye and Wang (2018). However, the work focuses on the task of keyphrase generation instead of keyphrase extraction.

Algorithm 1: Joint Learning based on Self-Distillation (JLSD)

```
Input: Labeled documents $\{(D_1, \mathbf{y}_1), ..., (D_n, \mathbf{y}_n)\}$ and unlabeled documents $\{\widetilde{D_1}, \widetilde{D_2}, ..., \widetilde{D_m}\}$
Train a teacher model using labeled documents.
Initialize a student model with same architecture and parameters as the teacher.
for $i = 1 \ \ldots \ T$ do
    Sample a batch of labeled documents $L$ uniformly at random.
    Sample a batch of unlabeled documents $\widetilde{U} = \{\widetilde{D_{i1}}, ..., \widetilde{D_{ik}}\}$ uniformly at random ($k = r|L|$).
    Use the teacher to generate (hard) pseudo labels for $\widetilde{U}$ to get $U = \{(\widetilde{D_{i1}}, \mathbf{y}_{i1}), ..., (\widetilde{D_{ik}}, \mathbf{y}_{ik})\}$.
    Use gradient descent to update the parameters of the student using examples in $L \cup U$.
    If (Student performance has improved) then Re-initialize the teacher using the student.
end
```

	Approach	Datasets	
		Inspec (F1)	SemEval-2017 (F1)
Previous Models	ELMo (Sahrawat et al., 2020)	0.568	0.504
	RoBERTA (Sahrawat et al., 2020)	0.595	0.508
	SciBERT (Sahrawat et al., 2020)	0.593	0.521
	BERT (Sahrawat et al., 2020)	0.591	0.522
SciBERT-based Models	SciBERT (Our implementation)	0.598 ± 0.006	0.544 ± 0.008
	SciBERT + JLSD	**0.605 ± 0.001**	0.546 ± 0.002
BERT-based Models	BERT (Our implementation)	0.596 ± 0.003	0.537 ± 0.003
	BERT + Simple Joint Training	0.582 ± 0.004	0.545 ± 0.008
	BERT + Simple Pretraining	0.588 ± 0.002	0.548 ± 0.004
	BERT + JLSD	0.601 ± 0.004	**0.554 ± 0.004**

Table 1: Overall results on the target datasets (means and standard deviations of our results are shown).

Metrics	Our Models				Unsupervised Ranking-based Models						
	BERT + JLSD	BERT	SciBERT + JLSD	SciBERT	SIF	Embed	RVA	Position	Topic	Single	YAKE
F1@5	42.06	41.97	**43.32**	42.39	29.11	27.20	21.91	25.19	22.76	24.69	15.73
F1@10	57.52	57.15	**58.05**	57.25	38.80	34.98	30.13	31.53	27.30	33.19	19.07
F1@15	66.05	66.01	**66.88**	66.13	39.59	36.45	33.22	33.30	28.42	35.42	20.17

Table 2: Comparison of our models and unsupervised models (F1@5, F1@10, and F1@15 test scores on the Inspec dataset are reported). To save space, standard deviations of our results are not shown.

4 Experiments and Results

Data and Experiments Setup In this work, we experimented with two target datasets: **Inspec** and **SemEval-2017**. The **Inspec** dataset (Hulth, 2003) has 1000/500/500 abstracts of scientific articles for the train/dev/test split. The **SemEval-2017** dataset (Augenstein et al., 2017) has 350/50/100 scientific articles for the train/dev/test split. In our experiments, we use the **KP20k** dataset (Meng et al., 2017) as the source dataset, because it contains more than 500,000 articles collected from various online digital libraries. Even though each article in the dataset has author-provided keyphrases, our proposed method does not require such supervised signals. We implemented two baseline models (Section 2) with different pre-trained contextual embeddings: BERT (base-cased)[1] and SciBERT (scivocab-cased)[2] (Wolf et al., 2019). From this point, we will refer to baseline models trained only on labeled data as [BERT] and [SciBERT]. We refer to models trained using our joint learning approach as [BERT + JLSD] and [SciBERT + JLSD]. For each variant, two different learning rates are used, one for the lower pretrained Transformer encoder and one for the upper layers. The optimal hyperparameter values are variant-specific, and we experimented with the following range of possible values: $\{4, 8, 16\}$ for batch size, {2e-5, 3e-5, 4e-5, 5e-5} for lower learning rate, {1e-4, 2e-4, 5e-4, 1e-3, 5e-3} for upper learning rate, and $\{25, 50, 75, 100, 125\}$ for number of training epochs. For JLSD, we experimented with various values for r among $\{0.25, 0.5, 1, 1.5, 2, 4\}$. We did hyper-parameter tuning using the provided dev sets.

Comparison with Previous Supervised Methods We first compare the performance of our models with the supervised models proposed by Sahrawat et al. (2020), as they have recently achieved state-of-the-art results on the two target datasets. Our baseline models are similar to the models proposed by Sahrawat et al. (2020), each consists of a contextualized embedding layer followed by a BiLSTM-CRF structure. Table 1 presents the overall results on the Inspec and SemEval-2017 datasets. Results of our models are averaged over three random seeds. When calculating the F1 scores, we consider only extractive keyphrases that are present in the documents. We see that our own implementation of the [BERT] and [SciBERT] variants achieve better results than reported by Sahrawat et al. (2020). Also, by applying our proposed joint learning method (JLSD), we can consistently improve the performance of the baseline variants. For example, the [SciBERT + JLSD] variant achieves new state-of-the-art performance

[1] https://huggingface.co/bert-base-cased

[2] https://huggingface.co/allenai/scibert_scivocab_cased

on the Inspec dataset, while the [BERT + JLSD] also outperforms previous methods on the SemEval-2017 dataset. These results demonstrate the effectiveness of our joint learning approach.

Comparison with Previous Unsupervised Methods We also compare our models with previous state-of-the-art unsupervised approaches, including SIFRank (Sun et al., 2020), EmbedRank (Bennani-Smires et al., 2018), RVA (Papagiannopoulou and Tsoumakas, 2018), PositionRank (Florescu and Caragea, 2017a), TopicRank (Bougouin et al., 2013), SingleRank (Wan and Xiao, 2008), and YAKE (Campos et al., 2018). Table 2 presents the comparison on the Inspec dataset. In this case, since the unsupervised methods are ranking-based methods, the performances are evaluated in terms of F1-measure when a fixed number of top keyphrases are extracted (i.e., F1@5, F1@10, and F1@15 measures are used). We see that our baseline models [BERT] and [SciBERT] already outperform previous unsupervised methods by a large margin. This agrees with previous studies, suggesting that in the presence of sufficient labeled data, supervised methods typically perform better than unsupervised methods (Kim et al., 2013; Caragea et al., 2014). Also, by applying JLSD, we further increase the gap with previous unsupervised methods. Note that, by default, our baseline model is a sequence labeling model (Section 2). Therefore, to compare our models with previous unsupervised ranking methods, we need to convert our models into ranking models by deriving a way to compute confidence scores for predicted keyphrases. In order to do so, we first calculate marginal probabilities after the CRF layer and simply make an independence assumption to compute the probability for a predicted keyphrase.

Comparison with Other Transfer Learning Techniques Finally, we compare our joint learning approach to other popular transfer learning techniques, including *simple pretraining* and *simple joint training*. Despite their simplicity, these strategies have been shown to be effective for a wide range of tasks (Min et al., 2017; Lai et al., 2018a; Yoon et al., 2019; Lai et al., 2019; Langenfeld et al., 2019). In *simple pretraining*, we first train a baseline model on the scientific articles in the source dataset. After that, we simply finetune the same model on a target dataset (i.e., Inspec or SemEval-2017). In *simple joint training*, we train a baseline model using examples from both the source dataset and the target dataset at the same time. Before each new training epoch, we sample a new set of examples from the source dataset and add them to the pool of examples of the target dataset. Different from our proposed approach, these two transfer learning techniques are only applicable to situations where the source dataset is labeled. In this case, since each article in the source dataset (i.e., KP20k) contains author-provided keyphrases, we use these keyphrases as the supervised signals. From the Table 1, we see that our variant [BERT + JLSD] outperforms both variants [BERT + Simple Joint Training] and [BERT + Simple Pretraining], even though [BERT + JLSD] does not require the source dataset to have any supervised signal. Furthermore, we see that the two simple transfer learning approaches even adversely decrease the performance of the BERT baseline model on the Inspec dataset.

5 Conclusion

In this work, we propose a novel joint learning approach based on self-distillation. Experimental results show that our approach consistently improves the performance of baseline models. Our best models even achieve new state-of-the-art results on two public benchmarks (Inspec and SemEval-2017). In future work, we plan to explore how to extend our method to other tasks (Brixey et al., 2018) and other languages. We will also investigate how our method can be used in few-shot settings.

References

Rabah Alzaidy, Cornelia Caragea, and C Lee Giles. 2019. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The world wide web conference*, pages 2551–2557.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada, August. Association for Computational Linguistics.

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

Florian Boudin and Emmanuel Morin. 2013. Keyphrase extraction for n-best reranking in multi-sentence compression. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 298–305, Atlanta, Georgia, June. Association for Computational Linguistics.

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

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

Jacqueline Brixey, Ramesh Manuvinakurike, Nham Le, Tuan Lai, Walter Chang, and Trung Bui. 2018. A system for automated image editing from natural language commands. *arXiv preprint arXiv:1812.01083*.

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

Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 2014. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1435–1446, Doha, Qatar, October. Association for Computational Linguistics.

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

Ernesto D'Avanzo and Bernado Magnini. 2005. A keyphrase-based approach to summarization: the lake system at duc-2005. In *Proceedings of DUC*.

Samhaa R El-Beltagy and Ahmed Rafea. 2009. Kp-miner: A keyphrase extraction system for english and arabic documents. *Information Systems*, 34(1):132–144.

Corina Florescu and Cornelia Caragea. 2017a. A position-biased pagerank algorithm for keyphrase extraction. In *AAAI*.

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

Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. *arXiv preprint arXiv:1805.04770*.

Sujatha Das Gollapalli, Xiao-Li Li, and Peng Yang. 2017. Incorporating expert knowledge into keyphrase extraction. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *EMNLP*.

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, July. Association for Computational Linguistics.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2013. Automatic keyphrase extraction from scientific articles. *Language resources and evaluation*, 47(3):723–742.

Tuan Lai, Trung Bui, Nedim Lipka, and Sheng Li. 2018a. Supervised transfer learning for product information question answering. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1109–1114. IEEE.

Tuan Manh Lai, Trung Bui, and Sheng Li. 2018b. A review on deep learning techniques applied to answer selection. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2132–2144, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Tuan Lai, Quan Hung Tran, Trung Bui, and Daisuke Kihara. 2019. A gated self-attention memory network for answer selection. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5953–5959, Hong Kong, China, November. Association for Computational Linguistics.

Tuan Manh Lai, Quan Hung Tran, Trung Bui, and Daisuke Kihara. 2020. A simple but effective bert model for dialog state tracking on resource-limited systems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8034–8038. IEEE.

Florent Langenfeld, Apostolos Axenopoulos, Halim Benhabiles, Petros Daras, Andrea Giachetti, Xusi Han, Karim Hammoudi, Daisuke Kihara, Tuan M Lai, Haiguang Liu, et al. 2019. Protein shape retrieval contest.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Marina Litvak and Mark Last. 2008. Graph-based keyword extraction for single-document summarization. In *Coling 2008: Proceedings of the workshop Multi-source Multilingual Information Extraction and Summarization*, pages 17–24, Manchester, UK, August. Coling 2008 Organizing Committee.

Debanjan Mahata, John Kuriakose, Rajiv Ratn Shah, and Roger Zimmermann. 2018. Key2Vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 634–639, New Orleans, Louisiana, June. Association for Computational Linguistics.

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

Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. 2017. Question answering through transfer learning from large fine-grained supervision data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 510–517, Vancouver, Canada, July. Association for Computational Linguistics.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2018. Local word vectors guiding keyphrase extraction. *Information Processing & Management*, 54(6):888–902.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Dhruva Sahrawat, Debanjan Mahata, Haimin Zhang, Mayank Kulkarni, Agniv Sharma, Rakesh Gosangi, Amanda Stent, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2020. Keyphrase extraction as sequence labeling using contextualized embeddings. In *European Conference on Information Retrieval*, pages 328–335. Springer.

Lucas Sterckx, Cornelia Caragea, Thomas Demeester, and Chris Develder. 2016. Supervised keyphrase extraction as positive unlabeled learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1924–1929, Austin, Texas, November. Association for Computational Linguistics.

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

Quan Hung Tran, Tuan Lai, Gholamreza Haffari, Ingrid Zukerman, Trung Bui, and Hung Bui. 2018. The context-dependent additive recurrent neural net. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1274–1283, New Orleans, Louisiana, June. Association for Computational Linguistics.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Hai Ye and L. Wang. 2018. Semi-supervised learning for neural keyphrase generation. In *EMNLP*.

Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2019. A compare-aggregate model with latent clustering for answer selection. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2093–2096.

Yuxiang Zhang, Yaocheng Chang, Xiaoqing Liu, Sujatha Das Gollapalli, Xiaoli Li, and Chunjing Xiao. 2017. Mike: keyphrase extraction by integrating multidimensional information. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1349–1358.

Xun Zhu, Chen Lyu, Donghong Ji, Han Liao, and Fei Li. 2020. Deep neural model with self-training for scientific keyphrase extraction. *Plos one*, 15(5):e0232547.