Keyphrase Extraction as Sequence Labeling Using Contextualized Embeddings

Dhruva Sahrawat[1], Debanjan Mahata[2,3](✉), Haimin Zhang[3],
Mayank Kulkarni[3], Agniv Sharma[2], Rakesh Gosangi[3], Amanda Stent[3],
Yaman Kumar[4], Rajiv Ratn Shah[2], and Roger Zimmermann[1]

[1] National University of Singapore, Singapore, Singapore
{dhruva,rogerz}@comp.nus.edu.sg
[2] Indraprastha Institute of Information Technology, New Delhi, India
rajivratn@iiitd.ac.in, agnivsharma96@gmail.com
[3] Bloomberg, New York City, USA
{dmahata,hzhang449,rgosangi,astent}@bloomberg.net
[4] Adobe, New Delhi, India
ykumar@adobe.com

Abstract. In this paper, we formulate keyphrase extraction from scholarly articles as a sequence labeling task solved using a BiLSTM-CRF, where the words in the input text are represented using deep contextualized embeddings. We evaluate the proposed architecture using both contextualized and fixed word embedding models on three different benchmark datasets, and compare with existing popular unsupervised and supervised techniques. Our results quantify the benefits of: (a) using contextualized embeddings over fixed word embeddings; (b) using a BiLSTM-CRF architecture with contextualized word embeddings over fine-tuning the contextualized embedding model directly; and (c) using domain-specific contextualized embeddings (SciBERT). Through error analysis, we also provide some insights into why particular models work better than the others. Lastly, we present a case study where we analyze different self-attention layers of the two best models (BERT and SciBERT) to better understand their predictions.

Keywords: Keyphrase extraction · Contextualized embeddings

1 Introduction

Keyphrase extraction is the process of selecting phrases that capture the most salient topics in a document [24]. They serve as an important piece of document metadata, often used in downstream tasks including information retrieval, document categorization, clustering and summarization. Classic techniques for keyphrase extraction involve a two stage approach [10]: (1) *candidate generation*, and (2) *pruning*. During the first stage, the document is processed to extract a set of candidate keyphrases. In the second stage, this candidate set is pruned to

J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12036, pp. 328–335, 2020.
https://doi.org/10.1007/978-3-030-45442-5_41

select the most salient candidate keyphrases, using either supervised or unsupervised techniques. In the supervised setting, pruning is formulated as a binary classification problem: *determine if a given candidate is a keyphrase.* In the unsupervised setting, pruning is treated as a *ranking problem*, where the candidates are ranked based on some measure of *importance*.

Challenges. Researchers typically employ a combination of different techniques for candidate generation such as extracting named entities, finding noun phrases that adhere to pre-defined lexical patterns [3], or extracting n-grams that appear in an external knowledge base like Wikipedia [9]. The candidates are further cleaned up using stop word lists or gazetteers. Errors in any of these techniques reduces the quality of candidate keyphrases. For example, if a named entity is not identified, it misses out on being considered as a keyphrase; if there are errors in part of speech tagging, extracted noun phrases might be incomplete. Also, since candidate generation involves a combination of heuristics with specific parameters, thresholds, and external resources, it is hard to be reproduced or migrated to new domains.

Motivation. Recently, researchers have started to approach keyphrase extraction as a sequence labeling task [1,8]. This formulation completely bypasses the candidate generation stage and provides a unified approach to keyphrase extraction. Unlike binary classification where each keyphrase is classified independently, sequence labeling finds an optimal assignment of keyphrase labels for the entire document. Sequence labeling allows to capture long-term semantic dependencies in the document.

There have been significant advances in deep contextual language models [7,21]. These models can take an input text and provide contextual embeddings for each token for use in downstream architectures. They have been shown to achieve state-of-the-art results for many different NLP tasks. More recent works [4,16] have shown that contextual embedding models trained on domain-specific corpora can outperform general purpose models.

Contributions. Despite all the developments, to the best of our knowledge, there hasn't been any work on the use of contextual embeddings for keyphrase extraction. We expect that, as with other NLP tasks, keyphrase extraction can benefit from contextual embeddings. We also posit that domain-specific language models may further help improve performance. To explore these hypotheses, in this paper, we approach keyphrase extraction as a sequence labeling task solved using a BiLSTM-CRF [11], where the underlying words are represented using various contextual embedding architectures. Following are the main contributions of this paper:

- We quantify the benefits of using deep contextual embedding models for sequence-labeling-based keyphrase extraction over using fixed word embeddings.
- We demonstrate the benefits of using a BiLSTM-CRF architecture with contextualized word embeddings over fine-tuning the contextualized word embedding model for keyphrase extraction.
- We demonstrate improvements using contextualized embeddings trained on a large corpus of in-genre text (SciBERT) over ones trained on generic text.

- We perform a robust set of experiments on three benchmark datasets, achieving state-of-the-art results and provide insights into the working of the different self-attention layers of our top-performing models.

2 Methodology

We approach the problem of automated keyphrase extraction from scholarly articles as a sequence labeling task, which can be formally stated as: Let $d = \{w_1, w_2, ..., w_n\}$ be an input text, where w_t represents the t^{th} token. Assign each w_t in the document one of three class labels $Y = \{k_B, k_I, k_O\}$, where k_B denotes that w_t marks the beginning of a keyphrase, k_I means that w_t is inside a keyphrase, and k_O indicates that w_t is not part of a keyphrase.

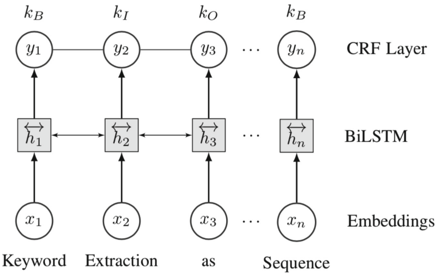


Fig. 1. BiLSTM-CRF architecture

In this paper, we employ a BiLSTM-CRF architecture (Fig. 1) to solve this sequence labeling problem. We map each token w_t in the input text to a fixed-size dense vector x_t. We then use a BiLSTM to encode sequential relations between the word representations. We then apply an affine transformation to map the output from the BiLSTM to the class space. The score outputs from the BiLSTM serve as input to a CRF [15] layer. In a CRF, the likelihood for a labeling sequence is generated by exponentiating the scores and normalizing over all possible output label sequences.

3 Experiments

Datasets. We ran our experiments on three different publicly available keyphrase datasets: Inspec [12], SemEval-2010 [14] (SE-2010), and SemEval-2017 [2] (SE-2017). Inspec consists of abstracts from 2000 scientific articles (train: 1000, validation: 500, and test: 500) where abstract is accompanied by both abstractive, i.e., not present in the documents, and extractive keyphrases. SE-2010 consists of 284 full length ACM articles (train: 144, validation: 40, and test: 100) containing both abstractive and extractive keyphrases. SE-2017 consists of 500 open access articles (train: 350, validation: 50, and test: 100) with location spans for keyphrases, i.e. all keyphrases are extractive.

Because we are modeling keyphrase extraction as a sequence labeling task, we only consider extractive keyphrases for our experiments. For Inspec and SE-2010, we identified the location spans for each extractive keyphrase. For SE-2010 and SE-2017, we only considered the abstract and discarded the remaining text due to memory constraints during inference with the contextual embedding models.

All the tokens in the datasets[1] were tagged using the B-I-O tagging scheme described in the previous section.

Experimental Settings. One of the main aims of this work is to study the effectiveness of contextual embeddings in keyphrase extraction. To this end, we use the BiLSTM-CRF with seven different pre-trained contextual embeddings: BERT [7] (small-cased, small-uncased, large-cased, large-uncased), SciBERT [4] (basevocab-cased, basevocab-uncased, scivocab-cased, scivocab-uncased), OpenAI GPT [22], ELMo [21], RoBERTa [17] (base, large), Transformer XL [5], and OpenAI GPT-2 [23] (small, medium). We also use 300 dimensional fixed embeddings from Glove [20], Word2Vec [19], and FastText [13] (common-crawl, wiki-news). We also compare the proposed architecture against four popular baselines: SGRank [6], SingleRank [26], Textrank [18], and KEA [27].

To train the BiLSTM-CRF models, we use stochastic gradient descent with Nesterov momentum in batched mode. The learning rate was set to 0.05 and the models were trained for a total of 100 epochs with patience value of 4 and annealing factor of 0.5. The hidden layers in the BiLSTM models were set to 128 units and word dropout set to 0.05. During inference, we run the model on a given abstract and identify keyphrases as all sequences of class labels that begin with the tag k_B followed by zero or more tokens tagged k_I. As in previous studies [14], we use F1-measure to compare different models. For each embedding model we report results for the best performing variant of that model (e.g. cased vs uncased) on each dataset.

4 Results

Of the ten embedding architectures, BERT or BERT-based models consistently obtained the best performance across all datasets (see Table 1). This was expected considering that BERT uses bidirectional pre-training which is more powerful. SciBERT was consistently one of the top performing models and was significantly better than any of the other models on SemEval-2010. Further analysis of the results shows that SciBERT was more accurate than other models in capturing keyphrases that contained scientific terms such as chemical names (e.g. Magnesium, Hydrozincite), software projects (e.g. HemeLB), and abbreviations (e.g. DSP, SIMLIB). SciBERT was also more accurate with keyphrases containing more than three tokens.

Table 1. Embedding models comparison (F1-score)

	Inspec	SE-2010	SE-2017
SciBERT	0.593	**0.357**	0.521
BERT	0.591	0.330	**0.522**
ELMo	0.568	0.225	0.504
Transformer-XL	0.521	0.222	0.445
OpenAI-GPT	0.523	0.235	0.439
OpenAI-GPT2	0.531	0.240	0.439
RoBERTa	**0.595**	0.278	0.508
Glove	0.457	0.111	0.345
FastText	0.524	0.225	0.426
Word2Vec	0.473	0.208	0.292
SGRank	0.271	0.229	0.211
SingleRank	0.123	0.142	0.155
TextRank	0.122	0.147	0.157
KEA	0.137	0.202	0.129

[1] https://github.com/midas-research/keyphrase-extraction-as-sequence-labeling-data.

Contextual embeddings outperformed their fixed counterparts for most of the experimental scenarios. The only exception was on SemEval-2010 where FastText outperformed Transformer-XL. Of the three fixed embedding models studied in this paper, FastText obtained the best performance across all datasets. Our model significantly outperforms all the four baseline methods for all three datasets irrespective of the embeddings used.

Contextualized embedding models can either serve as numerical representations of words for downstream architectures or they can be fine-tuned to be optimized for a specific task. Fine-tuning typically involves adding an untrained layer at the end and then optimizing the layer weights for the task-specific objective. We fine-tuned our best-performing contextualized embedding models (BERT and SciBERT) for each dataset and compared with the performance of the corresponding BiLSTM-CRF and BiLSTM models.

Both BiLSTM and BiLSTM-CRF models outperform fine-tuning across all datasets (see Table 2). We expect this might be due to the small sizes of the datasets on which the models were fine-tuned. The addition of the CRF layer improved the performance for all datasets. Analysis of results on the SemEval-2017 data shows that the CRF layer is more effective in capturing keyphrases that include prepositions (e.g. ‘of’), conjunctions (e.g. ‘and’), and articles (e.g. ‘the’). We also observed that the CRF layer is more accurate with longer keyphrases (more than two tokens).

Case Study: Attention analysis is used to understand if self-attention patterns in the layers of BERT and SciBERT provide any insight into the linguistic properties learned by the models. We present a case study of attention analysis for keyphrase extraction on a randomly chosen abstract from SemEval2017.

Table 2. Fine-tuning vs Pretrained (F1-score)

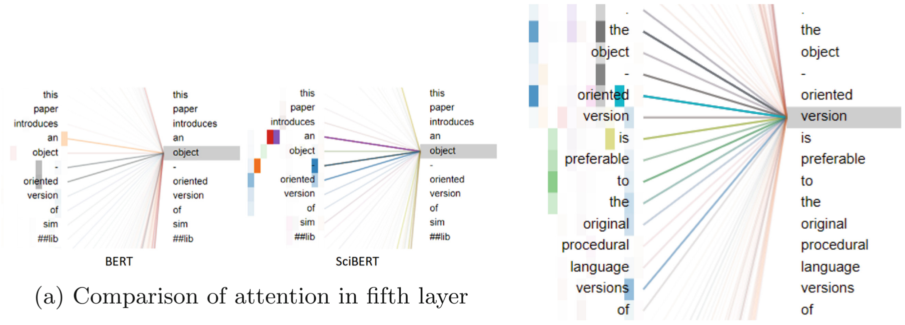
	BERT			SciBERT		
	Inspec	SE-2010	SE-2017	Inspec	SE-2010	SE-2017
Fine-tuning	0.474	0.236	0.270	0.488	0.268	0.339
BiLSTM-CRF	**0.591**	**0.330**	**0.522**	**0.593**	**0.357**	**0.521**
BiLSTM	0.501	0.295	0.472	0.536	0.301	0.455

Table 3. SciBERT vs BERT: keyphrase identification

SciBERT	BERT
An object-oriented version of SIMLIB -LRB- a simple simulation package -RRB- This paper introduces an object-oriented version of SIMLIB -LRB- an easy-to-understand discrete-event simulation package -RRB- . The object-oriented version is preferable to the original procedural language versions of SIMLIB in that it is easier to understand and teach simulation from an object point of view. A single-server queue simulation is demonstrated using the object-oriented SIMLIB	An object-oriented version of SIMLIB -LRB- a simple simulation package -RRB- This paper introduces an object-oriented version of SIMLIB -LRB- an easy-to-understand discrete-event simulation package -RRB- . The object-oriented version is preferable to the original procedural language versions of SIMLIB in that it is easier to understand and teach simulation from an object point of view. A single-server queue simulation is demonstrated using the object-oriented SIMLIB

Table 3 presents the classification results on this abstract from the BERT and SciBERT models; true positives in blue and false negatives in red. Using BertViz [25] we analyzed the aggregated attention of all 12 layers of both the models. We observed that keyphrase tokens (k_B and k_I) typically tend to pay most attention towards other keyphrase tokens. Contrarily, non-keyphrase tokens (k_O) usually pay uniform attention to their surrounding tokens. We found that both BERT and SciBERT exhibit similar attention patterns in the initial and final layers but they vary significantly in the middle layers. For example, the left figure in Table 4 compares the attention patterns in the fifth layer of both models. In SciBERT, the token 'object' is very strongly linked to other tokens from its keyphrase but the attentions are comparably weaker for BERT.

Table 4. Self-attention maps of layers in BERT and SciBERT



(a) Comparison of attention in fifth layer

(b) Attention towards similar tokens in SciBERT

We also observed that keyphrase tokens paid strong attention to similar tokens from other keyphrases. As shown in the right figure in Table 4, the token 'version' from 'object-oriented version' pays strong attention to 'versions' from 'procedural language versions'. This is a possible reason for both models failing to identify the third mention of 'object-oriented version' in the abstract as a keyphrase. We observed similar patterns in many other documents and we plan to quantify this analysis in future.

5 Conclusions

In this paper, we formulate keyphrase extraction as a sequence labeling task solved using BiLSTM-CRFs, where the underlying words are represented using various contextualized embeddings. We quantify the benefits of this architecture over direct fine tuning of the embedding models. We demonstrate how contextual embeddings significantly outperform their fixed counterparts in keyphrase extraction.

References

1. Alzaidy, R., Caragea, C., Giles, C.L.: Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents. In: Proceedings of WWW (2019)
2. Augenstein, I., Das, M., Riedel, S., Vikraman, L., McCallum, A.: SemEval 2017 task 10: ScienceIE-extracting keyphrases and relations from scientific publications. In: Proceedings of SemEval (2017)
3. Barker, K., Cornacchia, N.: Using noun phrase heads to extract document keyphrases. In: Hamilton, H.J. (ed.) AI 2000. LNCS (LNAI), vol. 1822, pp. 40–52. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45486-1_4
4. Beltagy, I., Cohan, A., Lo, K.: Scibert: pretrained contextualized embeddings for scientific text. arXiv:1903.10676 (2019)
5. Dai, Z., et al.: Transformer-XL: attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)
6. Danesh, S., Sumner, T., Martin, J.H.: SGRank: combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. In: Proceedings of Joint Conference on Lexical and Computational Semantics (2015)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT (2019)
8. Gollapalli, S.D., Li, X.l.: Keyphrase extraction using sequential labeling. arXiv preprint arXiv:1608.00329 (2016)
9. Grineva, M., Grinev, M., Lizorkin, D.: Extracting key terms from noisy and multitheme documents. In: Proceedings of WWW (2009)
10. Hasan, K.S., Ng, V.: Automatic keyphrase extraction: a survey of the state of the art. In: Proceedings of ACL (2014)
11. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. CoRR abs/1508.01991 (2015). http://arxiv.org/abs/1508.01991
12. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of EMNLP (2003)
13. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of EACL (2017)
14. Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T.: SemEval-2010 task 5: automatic keyphrase extraction from scientific articles. In: Proceedings of SemEval (2010)
15. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of ICML (2001)
16. Lee, J., et al.: BioBERT: pre-trained biomedical language representation model for biomedical text mining. arXiv preprint arXiv:1901.08746 (2019)
17. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
18. Mihalcea, R., Tarau, P.: TextRank: bringing order into text. In: Proceedings of EMNLP (2004)
19. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
20. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of EMNLP (2014)
21. Peters, M.E., et al.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)

22. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018). https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf
23. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog, February 2019
24. Turney, P.D.: Learning to extract keyphrases from text. arXiv preprint cs/0212013 (2002)
25. Vig, J.: A multiscale visualization of attention in the transformer model. arXiv preprint arXiv:1906.05714 (2019)
26. Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: Proceedings of AAAI (2008)
27. Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G.: Kea: practical automated keyphrase extraction. In: Design and Usability of Digital Libraries: Case Studies in the Asia Pacific, pp. 129–152. IGI Global (2005)