2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI) | 978-1-6654-0358-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/BHI50953.2021.9508592

Uncertainty-based Self-training for Biomedical Keyphrase Extraction

1st Zelalem Gero
Department of Computer Science
Emory University
Atlanta, USA
zgero@emory.edu

2nd Joyce C. Ho
Department of Computer Science
Emory University
Atlanta, USA
joyce.c.ho@emory.edu

***Abstract*—To keep pace with the increased generation and digitization of documents, automated methods that can improve search, discovery and mining of the vast body of literature are essential. Keyphrases provide a concise representation by identifying salient concepts in a document. Various supervised approaches model keyphrase extraction using local context to predict the label for each token and perform much better than the unsupervised counterparts. However, existing supervised datasets have limited annotated examples to train better deep learning models. In contrast, many domains have large amount of un-annotated data that can be leveraged to improve model performance in keyphrase extraction. We introduce a self-learning based model that incorporates uncertainty estimates to select instances from large-scale unlabeled data to augment the small labeled training set. Performance evaluation on a publicly available biomedical dataset demonstrates that our method improves performance of keyphrase extraction over state of the art models.**

Keywords: Keyphrase Extraction, Document Summarization, Biomedical text processing

I. Introduction

Keyphrase extraction is an important information extraction task that identifies single or multi-word linguistic units to concisely represent a document. The keyphrases also provide a brief summary of the document content. Keyphrases are widely used in variety of natural language processing (NLP) tasks such as document summarization [1], [2], text classification [3], and recommendation systems [4]. Existing keyphrase extraction methods either take a supervised or unsupervised approach. Common unsupervised approaches are graph-based ranking algorithms where each word is a node and edges connect words that co-occur within a specified window size. While unsupervised approaches are desirable for datasets with limited ground truth values, most such methods perform worse compared to the supervised counterparts [5].An example of a biomedical abstract with annotated keyphrases is shown in Fig **??**.

The supervised keyphrase extraction approaches use classification to label every token by using features such as part-of speech tags, term-frequency inverse document frequency (tf-idf), and the position of the token in the document. Recently, deep learning (DL) models have been employed for keyphrase extraction. Several works posed the problem as a sequence labeling task and applied long short term memory (LSTM) and conditional random fields (CRF) to tag each token in document as positive (is part of a keyphrase) or negative [6], [7]. This approach has the benefit of considering the whole document sequence when assigning labels instead of independently classifying each token to capture semantic dependencies among tokens in the entire document. While supervised DL-based methods achieve state of the art results in keyphrase extraction tasks, their performances lag behind other common NLP tasks such as text classification and Named Entity Recognition (NER) due to lack of large annotated datasets, especially in the scientific domain.

One way to overcome the lack of annotated data is to leverage the widely available scientific articles published online in a semi-supervised fashion. Even though a small percentage of these documents contain human annotated keyphrases, the unlabeled documents can be used to learn better representations. We propose to make use of these large scale unlabeled scientific articles by using self-training and uncertainty estimation to improve the performance of supervised keyphrase extraction.

Our model, based on the sequence labeling DL-model, is first trained on the small labeled dataset and then used to generate pseudo-labels (i.e., labels that are annotated by a trained model instead of a human annotation) for the unlabeled documents. As these labels are noisy, our model estimates the model uncertainty using Monte Carlo (MC) Dropout and then selects a subset of these pseduo-labeled data to retrain the classifier. The contributions of this work can be summarized as:

- A new uncertainy-based, self-training approach for keyphrase extraction model that uses unlabeled data for better performance.
- Introduction of Monte Carlo Dropout and model uncertainty estimation for pseudo-labeled document sample selection.
- Demonstration of the effectiveness of our approach on a publicly available biomedical keyphrase extraction dataset.

II. Related Work

Keyphrase extraction methods mainly take either supervised or unsupervised approach. Unsupervised approaches generate

978-1-6654-0358-0/21/$31.00 ©2021 IEEE

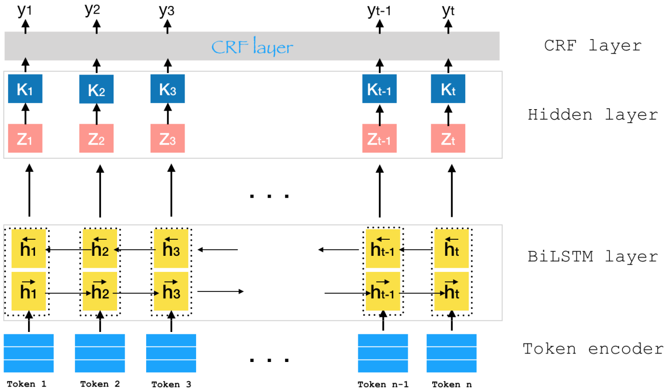

Fig. 1: A common baseline BiLSTM-CRF architecture for keyphrase extraction.

candidates and rank using features such as tf-idf and topic proportions [8], [9], graph based centrality measures [10]–[12], and topic modeling [10], [13].

For supervised keyphrase extraction models, this can be posed as a binary classification task [14]–[16] or as a ranking between candidates [17]. Candidates keys are extracted using statistical (e.g., number of occurrences, first occurrence of the key) and structural features (e.g., part of speech tags). DL-based models have also been used for keyphrase extraction. For example, a Recurrent Neural Network (RNN) based approach was used to identify keyphrases in Twitter data [18] and an attention-based neural network was used to extract keyphrases from scientific documents by retrieving additional information from other sentences within the same document [19]. Most notably, Al-Zaidy et al. employed a BiLSTM-CRF architecture to model keyphrase extraction as a sequence labelling task [14]. Sahrawat et al. [5] evaluate the effect of various pre-trained word embeddings on the BiLSTM-CRF architecture for several benchmark datasets.

Since DL-models require significant labeled data, self-training for keyphrase extraction has been explored recently [20], [21]. Even though their approaches show performance gains over baseline models, the uncertainty of the model is not incorporated which can lead to poor learning and noise propagation. We propose to incorporate the uncertainty of the the model during self-training for further improvements.

III. Methodology

The keyphrase extraction task is formulated as a sequence labelling task. Given a document $X = w_1, w_2, \cdots, w_t$ where w_i is the i^{th} token and t is the number of tokens in the document, we predict the labels $Y = \{k_B, k_I, k_O\}$ where k_B, k_I and k_O denote whether the token is the beginning of, part of, or not a part of a keyphrase, respectively. The baseline DL-model we employ is the commonly used BiLSTM-CRF architecture [5], [14], [20], [21] shown in Figure 1. We first briefly describe the BiLSTM-CRF model before introducing self-training and uncertainty estimation for keyphrase extraction.

A. BiLSTM-CRF Architecture

a) Token Emebedding: Each token, w_i, is represented by a low-dimensional vector representations x_i. Any pre-trained word embedding can be used such as Glove [22], word2vec [23], SciBERT [24] and BioBERT [25]. Contextualized embeddings such as SciBERT have been shown to provide better results [5].

b) BiLSTM Layer: A BiLSTM layer is used to encode each document into a local contextual representation. The BiLSTM generates two feature representations, $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$, for each x_i using a forward and backward LSTM, respectively. The two representations are concatenated and then passed to an affine transformation $k_t = W_a \overset{\leftrightarrow}{[} \overrightarrow{h_t}; \overleftarrow{h_t}]$.

c) CRF: Given the sequence of tokens, CRF produces a probability distribution over the output label sequence using the dependencies among the labels of the entire input sequence [26]. Given a transition matrix Γ where $\Gamma_{i,j}$ is the transition score from class y_{t-1} to y_t, the score of an output label sequence s is given by $s(s, y) = \sum_{t=1}^{n} \Gamma_{y_{t-1}, y_t} + K_t, y_t$. The overall likelihood score for a given sequence is then calculated by exponentiating the individual scores and normalizing over all possible output sequences.

B. Self-training and Uncertainty Estimation

Self-training is a semi-supervised approach which has state-of-the art performances across several applications [27]–[29]. Under the self-training paradigm, a teacher model is trained on a small amount of labeled data ($\mathcal{D}_l$) and used to generate pseudo-labels on unlabeled data ($\mathcal{D}_u$). A subset of the pseudo-labeled data is then combined with the labeled data to train a second model called a student model. The student then becomes the teacher and this process is repeated until convergence is achieved.

While several self-training keyphrases extraction models have been proposed [20], [21], they fail to consider the teacher uncertainty. These implementations only sample pseudo-labeled instances where the model confidence is high in a single pass. Predictive probabilities from a softmax output are erroneously taken as model confidence. Gal et al. [30] demonstrate that a model can be uncertain in its predictions even with a high softmax output. This can lead to poor learning and noise propagation through self-training on wrong pseudo-labels [31]. Moreover, selecting samples where the model is very confident may not improve the performance of the student model as these may already be correctly classified. However, selecting samples where the model is least confident can make it difficult to learn anything important. Mukherjee et al. [31] proposed to select examples based on the uncertainty of the teacher model to improve the self-training process by modeling a distribution over the parameters through Bayesian Neural Networks to reflect model uncertainty. Unfortunately, direct adoption of this framework is not straightforward as questions arise from the multiple pseudo-keyphrase annotations associated with each document.

Based on the promising results of using uncertainty to improve the self-training process, we introduce a new uncertainty-based self-training model for keyphrase extraction. For each sample in the unlabeled data ($\mathcal{D}_u$), we use Monte Carlo dropout [30] to simulate a Bayesian approximation to quantify the uncertainty associated with the teacher model f_t with corresponding model parameters W. This means that M forward passes are performed where stochastic dropouts are applied to each hidden layer ($\tilde{W}_m$) to approximate the model output as a random sample from the posterior distribution as in [31]. It is important to note that this process will create different pseudo keyphrases for each document since dropouts are activated during inference as well. Thus for each unlabeled sample x_u, there are M pseudo-labels for each token in the document, $y_1^*, \cdots, y_M^*$. The pseudo-labels are used to compute the stochastic mean and variance of x_u:

$$E(y) = \frac{1}{M} \sum_{m=1}^{M} y_m^*(x) \tag{1}$$

$$Var(y) \approx \frac{1}{M} \sum_{m=1}^{M} y_m^*(x)^\top y_m^*(x) - E(y)^\top E(y) \tag{2}$$

From these, model uncertainty is approximated by the summary of variance of the model outputs from the multiple forward passes. The uncertainty for a given unlabeled document is the mean of the uncertainties of the individual tokens. This gives us the pseudo-labels with their corresponding uncertainties $u_1, u_2, ..., u_m$ for each unlabeled document x_u. Pseudo-labeled samples with low uncertainty values are considered easier while high uncertainty valued samples are harder for the teacher model to predict. To enhance the student learning, we select samples with the average uncertainty value less than a threshold (we used 0.2 since this gives the best results on the validation set). This helps with selecting some samples where the teacher model is not very certain.

Algorithm 1 outlines our uncertainty-based self-training process for keyphrase extraction.

Algorithm 1: Pseudo-code for iterative self-training

```
Train $f_t$ teacher model with parameters $W$ on $\mathcal{D}_l$;
while not converged do
    for $x \in \mathcal{D}_u$ do
        for $m \in \{1, \cdots, M\}$ do
            $\tilde{W}_m \sim Dropout(W)$ ;
            $y_m^* = softmax\left(f^{(\tilde{W}_m)}(x)\right)$ ;
        end
        Calculate stochastic mean and variance of $x$ ;
    end
    Sample instances with uncertainty less than a given
      threshold ($\alpha$) ;
    Retrain model $W$ using the combined data ;
end
```

IV. Experiments

Datasets. We ran our experiment on a publicly available scientific keyphrase dataset: PubMed [32]. PubMed contains 2532 articles from PubMed Central Open Access Subset with at least 5 author-provided keyphrases. Since we use a sequence labeling formulation, the document/keyphrases data pairs are prepared such that each document is a sequence of word tokens, where the positive labels (k_B, k_I) are used if the word occurs in a keyphrase and a negative label (k_O) if it is not part of the keyphrase.

For the self-training based model, we use an unlabeled dataset. For the PubMed keyphrases, we utilize PubMed-Medline[1] which contains over 28 million abstracts of biomedical journals.

Experiment Settings. The baseline models are Bi-LSTM-CRF with two different word embeddings: 768-dimension SciBERT [24], and 768-dimension BioBERT [25].

For the PubMed dataset, we split into 80%, 10%, and 10% for training, validation and testing, respectively. The Bi-LSTM-CRF models are optimized during training using stochastic gradient descent with a learning rate 0.0001. Gradient clipping of 5.0 is used to prevent the gradient from overflows during back-propagation. In addition, we use dropout to avoid over-fitting. We evaluate the models using the F1 score on the test set using three different runs.

A. Evaluation Results

To quantify the performance benefits of self-training for keyphrase extraction, we have used two of the best performing pre-trained models commonly used: SciBert and BioBert. These pre-trained models already achieve state-of-the art performances in many downstream tasks. We fine-tuned the pre-trained models by adding a Bi-LSTM and CRF layers with small labeled data available. After fine-tuning on the small labeled data, we use the self-training module to keep sampling from the unlabeled set.

The performance comparison of the baselines and our model is shown in Table I. Since pre-trained models already use large amount of unlabeled data, it's usually cumbersome to squeeze out performance improvements. Our model shows significant performance gain on the PubMed dataset compared to the baselines. The improvement gained from our model is not as large on the INSPEC dataset compared to the other self-training based baselines. However, we still get improvements over pre-trained strong baselines showing the potential of self-training to improve down stream task fine-tuning.

For the sake of comparison with common unsupervised approaches, we have ranked our keyphrase tagging based on the model uncertainty. In Table II we show F1 scores when extracting 5,10, and 15 keyphrases from a document. The results on the PubMed dataset show that unsupervised methods lag way behind their supervised counterparts as our model performs magnitudes better.

V. Conclusion

In this paper, we proposed a new uncertainty-based self-training keyphrase extraction method that utilizes unlabeled

[1] https://www.nlm.nih.gov/databases/download/pubmed_medline.html

TABLE I: Comparison of model performance by fine-tuning pre-trained models

Model	F1 score
Bi-LSTM(SciBert) + CRF	0.765(±0.003)
Bi-LSTM(BioBert) + CRF	0.768(±0.003)
SciBERT + JLSD([20])	0.765(±0.003)
SciBERT_sahrawat([5]	0.766(±0.002)
Ours(BioBert + CRF+ Self)	**0.773**(±0.002)

TABLE II: Comparison of common unsupervised models and our model on PubMed dataset

	SingleRank	PositionRank	TopicRank	Ours
F1@5	15.2	18.3	26.4	**36.2**
F1@10	16.3	18.3	28.7	**54.3**
F1@15	19.2	20.9	29.2	**64.5**

data to augment small labeled training data. We introduce Monte Carlo dropout to approximate the model uncertainty for each pseudo-labeled document. The uncertainty is then used to sample specific documents to retrain the model using the combined data.This iterative Teacher-Student model training is performed until convergence is achieved. The empirical results on the two datasets showcase that self-training can provide an performance improvement, especially for PubMed where there is a significant unlabeled corpus.

Acknowledgements

This work is supported by funding from the National Institute of Health awards #5K01LM012924 and #1R01LM013323.

References

[1] S. K. Bharti and K. S. Babu, “Automatic keyword extraction for text summarization: A survey,” *arXiv preprint arXiv:1704.03242*, 2017.

[2] K. Sarkar, “A keyphrase-based approach to text summarization for english and bengali documents,” *International Journal of Technology Diffusion (IJTD)*, vol. 5, no. 2, pp. 28–38, 2014.

[3] F. Coenen, P. Leng, R. Sanderson, and Y. J. Wang, “Statistical identification of key phrases for text classification,” in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2007, pp. 838–853.

[4] N. Naw and E. E. Hlaing, “Relevant words extraction method for recommendation system,” *Bulletin of Electrical Engineering and Informatics*, vol. 2, no. 3, pp. 169–176, 2013.

[5] D. Sahrawat, D. Mahata, H. Zhang, M. Kulkarni, A. Sharma, R. Gosangi, A. Stent, Y. Kumar, R. R. Shah, and R. Zimmermann, “Keyphrase extraction as sequence labeling using contextualized embeddings,” in *Proc. of ECIR*, 2020, pp. 328–335.

[6] S. Thomaidou and M. Vazirgiannis, “Multiword keyword recommendation system for online advertising,” in *International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2011, Kaohsiung, Taiwan, 25-27 July 2011*. IEEE Computer Society, 2011, pp. 423–427. [Online]. Available: https://doi.org/10.1109/ASONAM.2011.70

[7] S. D. Gollapalli, X.-L. Li, and P. Yang, “Incorporating expert knowledge into keyphrase extraction,” in *Proc. of AAAI*, 2017, pp. 3180–3187.

[8] K. Barker and N. Cornacchia, “Using noun phrase heads to extract document keyphrases,” in *Conference of the canadian society for computational studies of intelligence*. Springer, 2000, pp. 40–52.

[9] Z. Liu, P. Li, Y. Zheng, and M. Sun, “Clustering to find exemplar terms for keyphrase extraction,” in *Proc. of EMNLP*. Singapore: Association for Computational Linguistics, 2009, pp. 257–266. [Online]. Available: https://www.aclweb.org/anthology/D09-1027

[10] A. Bougouin, F. Boudin, and B. Daille, “TopicRank: Graph-based topic ranking for keyphrase extraction,” in *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. Nagoya, Japan: Asian Federation of Natural Language Processing, 2013, pp. 543–551. [Online]. Available: https://www.aclweb.org/anthology/I13-1062

[11] M. Litvak and M. Last, “Graph-based keyword extraction for single-document summarization,” in *Coling 2008: Proceedings of the workshop Multi-source Multilingual Information Extraction and Summarization*, 2008, pp. 17–24.

[12] C. Florescu and C. Caragea, “PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents,” in *Proc. of ACL*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 1105–1115. [Online]. Available: https://www.aclweb.org/anthology/P17-1102

[13] F. Liu, D. Pennell, F. Liu, and Y. Liu, “Unsupervised approaches for automatic keyword extraction using meeting transcripts,” in *Proc. of NAACL-HLT*, 2009, pp. 620–628.

[14] R. Alzaidy, C. Caragea, and C. L. Giles, “Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents,” in *Proc. of WWW*, 2019, pp. 2551–2557.

[15] P. D. Turney, “Learning algorithms for keyphrase extraction,” *Information retrieval*, vol. 2, no. 4, pp. 303–336, 2000.

[16] ——, “Learning to extract keyphrases from text,” *arXiv preprint cs/0212013*, 2002.

[17] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning, “Kea: Practical automated keyphrase extraction,” in *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*. IGI Global, 2005, pp. 129–152.

[18] Q. Zhang, Y. Wang, Y. Gong, and X.-J. Huang, “Keyphrase extraction using deep recurrent neural networks on Twitter,” in *Proc. of EMNLP*, 2016, pp. 836–845.

[19] T. Y. S. S. Santosh, D. K. Sanyal, P. K. Bhowmick, and P. P. Das, “DAKE: Document-level attention for keyphrase extraction,” in *Proc. of ECIR*. Springer, 2020, pp. 392–401.

[20] T. M. Lai, T. Bui, D. S. Kim, and Q. H. Tran, “A joint learning approach based on self-distillation for keyphrase extraction from scientific documents,” in *Proc. of the 28th International Conference on Computational Linguistics*, 2020, pp. 649–656.

[21] X. Zhu, C. Lyu, D. Ji, H. Liao, and F. Li, “Deep neural model with self-training for scientific keyphrase extraction,” *Plos one*, vol. 15, no. 5, p. e0232547, 2020.

[22] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vectors for word representation,” in *Proc. of EMNLP*, 2014, pp. 1532–1543.

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proc. of NeurIPS*, 2013, pp. 3111–3119.

[24] I. Beltagy, K. Lo, and A. Cohan, “Scibert: A pretrained language model for scientific text,” in *Proc. of EMNLP*, 2019, pp. 3615–3620.

[25] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, “Biobert: a pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[26] J. Lafferty, A. McCallum, and F. C. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” 2001.

[27] C. Rosenberg, M. Hebert, and H. Schneiderman, “Semi-supervised self-training of object detection models,” 2005.

[28] J. Tanha, M. van Someren, and H. Afsarmanesh, “Semi-supervised self-training for decision tree classifiers,” *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 1, pp. 355–370, 2017.

[29] X. Li, Q. Sun, Y. Liu, S. Zheng, Q. Zhou, T.-S. Chua, and B. Schiele, “Learning to self-train for semi-supervised few-shot classification,” in *Proc. of NeurIPS*, 2019, pp. 10 276–10 286.

[30] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *Proc. of ICML*, 2016, pp. 1050–1059.

[31] S. Mukherjee and A. H. Awadallah, “Uncertainty-aware self-training for text classification with few labels,” in *Proc. of NeurIPS*, 2020, pp. 21 199–21 212.

[32] Z. Gero and J. C. Ho, “Namedkeys: Unsupervised keyphrase extraction for biomedical documents,” in *Proc. of BCB*, 2019, pp. 328–337.