PLOS ONE

RESEARCH ARTICLE

Deep neural model with self-training for scientific keyphrase extraction

Xun Zhu[1,2], **Chen Lyu**[3,4]*, **Donghong Ji**[1]*, **Han Liao**[2], **Fei Li**[1]

1 Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, Hubei, China, **2** School of Mathematics and Computer Science, Jianghan University, Wuhan, Hubei, China, **3** Laboratory of Language and Artificial Intelligence, Guangdong University of Foreign Studies, Guangzhou, Guangdong, China, **4** Collaborative Innovation Center for Language Research and Services, Guangdong University of Foreign Studies, Guangzhou, Guangdong, China

* lvchen1989@whu.edu.cn (CL); dhji@whu.edu.cn (DJ)

Citation: Zhu X, Lyu C, Ji D, Liao H, Li F (2020) Deep neural model with self-training for scientific keyphrase extraction. PLoS ONE 15(5): e0232547. https://doi.org/10.1371/journal.pone.0232547

Editor: Weinan Zhang, National University of Singapore, SINGAPORE

Received: December 27, 2019

Accepted: April 16, 2020

Published: May 15, 2020

Data Availability Statement: The ScienceIE corpus can be downloaded at: https://scienceie.github.io/resources.html#. The ACL keyphrase corpus can be downloaded at: https://nlp.stanford.edu/pubs/FTDDataset_v1.txt.

Funding: This work was supported in part by the National Natural Science Foundation of China (No. 61772378), the Social Science Foundation of Ministry of Education of China (No. 18JZD015, 19YJCZH114), the Natural Science Foundation of Hubei Province (No. 2012FFA088), the Special Innovation Project of Guangdong Education Department (No. 2018KTSCX059), and the

Abstract

Scientific information extraction is a crucial step for understanding scientific publications. In this paper, we focus on scientific keyphrase extraction, which aims to identify keyphrases from scientific articles and classify them into predefined categories. We present a neural network based approach for this task, which employs the bidirectional long short-memory (LSTM) to represent the sentences in the article. On top of the bidirectional LSTM layer in our neural model, conditional random field (CRF) is used to predict the label sequence for the whole sentence. Considering the expensive annotated data for supervised learning methods, we introduce self-training method into our neural model to leverage the unlabeled articles. Experimental results on the ScienceIE corpus and ACL keyphrase corpus show that our neural model achieves promising performance without any hand-designed features and external knowledge resources. Furthermore, it efficiently incorporates the unlabeled data and achieve competitive performance compared with previous state-of-the-art systems.

Introduction

With the explosive increase of scientific publications, it is important for users to better understand the key ideas of the articles. Keyphrases are usually regarded as phrases that represent the salient concepts of a document [1], and provide users with valuable information. The scientific keyphrases identification and classification is motivated by the increasing demand for efficiently finding relevant scientific publications and automatically understanding the key information of those publications, and it has received much academic interest over the past years [2–6]. Furthermore, it is also an important prerequisite task for downstream applications, such as summarization, information retrieval and question-answering.

Scientific information extraction, including keyphrase and semantic relation extraction, is a curial step for understanding scientific publications. SemEval 2017 has organized a shared task on scientific information extraction (ScienceIE) [6]. The benchmark dataset consists of

National Key Research and Development Program of China (No. 2017YFC1200500). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

scientific articles in the Computer Science, Material Sciences and Physics domains, and the keyphrases in this dataset are annotated with three categories: TASK, PROCESS and MATERIAL. One annotated example in this dataset is illustrated in Fig 1. Gupta and Manning (2011) [7] introduced a human-annotated dataset containing abstracts from the ACL anthology. The keyphrases with three categories (DOMAIN, TECHNIQUE and FOCUS) are annotated in this corpus. These annotated datasets allow us to employ supervised machine learning methods for scientific keyphrase extraction.

Most studies have been conducted on keyphrase extraction [8]. On one hand, unsupervised methods, such as TF-IDF [9] based ranking method, achieve comparable performance in this task [10, 11]. On the other hand, various supervised learning methods, such as support vector machines (SVMs), are used to identify keyphrases [6, 12]. However, these systems need to pay much attention on feature engineering efforts.

Recently, deep learning has been widely used in natural language processing (NLP) [13–16], and it brings hope to reduce manual feature engineering in various tasks. Compare with hand-designed features and traditional discrete feature representation, it provides a different way to automatically learn dense features representation for text, such as words, phrases and sentences. Our method follows this line and builds the neural model based on the bidirectional long short-memory (LSTM) and conditional random field (CRF). The input word representation of our model is computed based on the word embeddings, part-of-speech (POS) embeddings and dependency embeddings.

However, the success of these supervised learning methods relies on large amounts of annotated data. The number of training instances of this task is limited and it is expensive to annotate much more data for supervised training of neural network models. As we know, there are massive unannotated scientific publications, which are publicly available. In this paper, we introduce self-training methods to the neural model to take advantage of these unlabeled scientific articles. We first train the model with the original training data, and label the unannotated text using this model. The high confidence data with the predicted label is selected into the training set. Then we retrain the model using the new training data. The self-training method repeatedly perform the above process, and a number of data is selected into the training set to improve the performance.

We evaluate our models on the SemEval 2017 ScienceIE corpus and ACL keyphrase corpus. Standard evaluation demonstrates that our neural model can achieve promising performance for scientific keyphrase extraction without any hand-designed features and external knowledge resources. In addition, our model with self-training method can efficiently utilize unlabeled data, and achieve competitive performance compared with other state-of-the-art systems.

Domain: Material Science:

Text: Poor **[PROCESS oxidation]** behavior is the major barrier to the increased use of **[MATERIAL Ti-based alloys]** in high-temperature structural applications.

No.	Type	Start	End	Keyphrase
1	Process	5	14	oxidation
2	Material	69	84	Ti-based alloys

Fig 1. Keyphrases annotated in the ScienceIE corpus.

https://doi.org/10.1371/journal.pone.0232547.g001

Related work

Keyphrase extraction

Previous works on keyphrase extraction usually focus on documents in different domains, including news [17], scientific [4], meeting transcripts [10] and web text [18, 19]. The methods used for keyphrase extraction fall into two lines: supervised learning and unsupervised learning.

In the supervised learning research line, keyphrase extraction is formalized as a classification problem. These works first extract candidate phrases using some heuristic rules, and then train a classification model to predict whether a candidate phrase is a keyphrase or not. Different features have been used for this task [2, 20–22], including frequency features (e.g. TF-IDF), position features, structural features, syntactic features and external resource-based features.

In the unsupervised learning research line, it is usually formalized as a ranking problem. The kayphrases are usually ranked based on the TF-IDF [10, 23, 24] and term informativeness [25]. Besides the frequency information, more statistic and context information [26] has shown the importance in this task. Graph-based ranking is also widely used in unsupervised methods [17, 27]. It aims to build a graph and rank its nodes, which represent candidate keyphrases, according to their importance. Following this approach, topic information [28, 29], semantic information from knowledge bases [30, 31] and pretrained word embeddings [32, 33] have been incorporated into the graph-based ranking model to improve the performance.

Due to the lack of human-annotated corpus, previous work on scientific information extraction is limited. Gupta and Manning (2011) [7] introduced a dataset of scientific abstracts annotated with the three categories. Pattern-based bootstrapping approach is used to automatically extract these keyphrases. Tsai *et al.* (2013) [34] incorporated hand-designed features into unsupervised bootstrapping framework to improve the performance.

Scientific information extraction has attracted much attention in recent years, and it becomes the focus of SemEval 2017 Task10. This task includes three subtasks: keyphrase identification, keyhprase classification and relation extraction between keyphrases. It releases the ScienceIE dataset, and provides a benchmark to evaluate the system performance on this task. In this paper, we focus on kephrase extraction, including keyphrase identification and classification. Similar to named entity recognition (NER), we formalize keyphrase extraction as a sequence labelling problem.

Deep learning

Deep neural networks, especially recurrent neural networks (RNN) [35] and LSTM-RNN [36] have been successfully used for the sequence labelling task. Collobert *et al.* (2011) [13] proposed a unified neural network framework, and performed various sequence labelling tasks, including POS-tagging, NER and chunking, simultaneously. Huang *et al.* (2015) [37] used hand-crafted features with LSTMs to improve the NER performance. The bidirectional LSTM-RNN combined with CRFs have been widely used in NER [37–40], and achieve promising performance.

Word2Vec [41] and GloVe [42] are effective algorithms for learning word representations to capture syntactic and semantic features for words. Recently, pre-trained language models with context, such as ELMo [43], BERT [44] and XLNet [45], have shown great power in the semantic representation of text, and achieved excellent performance in various NLP applications.

In scientific keyphrase extraction subtask of SemEval 2017 Task 10, top three systems all used RNN-based methods [46, 47]. On the top of the RNN layer, CRFs [48], which jointly

Sentence 1:

This/O paper/O addresses/O the/O task/O of/O **named/B_Task entity/I_Task recognition/L_Task** (/O **NER/U_Task**)/O ,/O a/O subtask/O of/O **information/B_Task extraction/L_Task** ,/O using/O **conditional/B_Process random/I_Process fields/L_Process** (/O **CRF/U_Process**)/O ./O Our/O method/O is/O evaluated/O on/O the/O **ConLL-2003/B_Material NER/I__Material corpus/L_Material** ./O

Fig 2. One example using the BILUO label scheme. Sentence 1 comes from the ScienceIE corpus.

https://doi.org/10.1371/journal.pone.0232547.g002

model the label sequence, performed better compared to the softmax layer. Alzaidy *et al.* (2019) [49] explored a Bi-LSTM-CRF neural model, that captures long distance semantic information, for keyphrase extraction from scientific articles. Our work is related to the line and introduces self-training method to the neural model to leverage unlabeled scientific articles.

Methods

Label schemes for NER

We introduce the label scheme used for this task in this section. Following previous works on NER, we use the BILUO label scheme, which has been widely used for NER, for our task in this paper.

B and L are used to label the beginning and end word of the keyphrase. U indicates the one-word keyphrase. I and O are used to label the the inside and outside words of the keyphrase. Meanwhile, suffix is added to represent the category of the keyphrase, such as PROCESS, MATERIAL and TASK. One example using the BILUO label scheme is shown in Fig 2.

BLSTM-CRF model

Our method utilizes the bidirectional LSTM (BLSTM) and CRF for scientific keyphrase extraction. Fig 3 illustrates the BLSTM-CRF model used for this task. It first gets the word

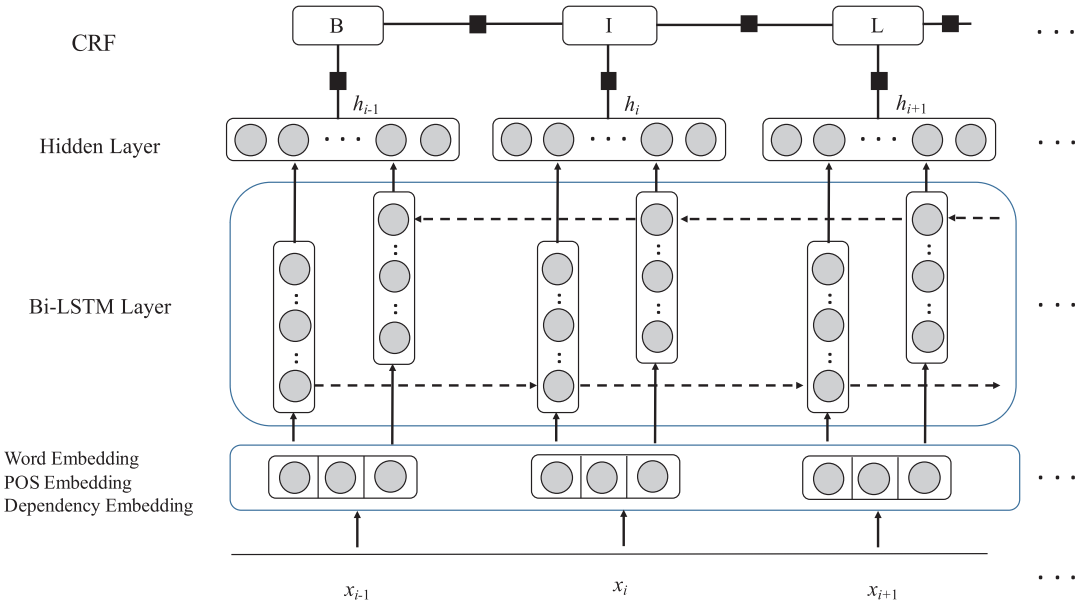

Fig 3. Overview of the BLSTM-CRF model.

https://doi.org/10.1371/journal.pone.0232547.g003

representation by concatenating word embeddings, POS embeddings and dependency embeddings. Then the Bi-LSTM layer takes the word representation as input and generate more complex features for the input sentence. Finally, CRF is added to predict the label sequence for the sentence. The BILUO label scheme mentioned in the above section is used to form the label sequence.

Word embeddings, which are trained from large scale raw corpus, can capture semantic information of the word. They are commonly used in the neural network models and give a continuous feature vector to represent the word. Given an input sentence $s = \{w_1, w_2, \ldots, w_n\}$, the word embedding lookup table function is used to get the word embedding e_i for each word w_i.

Not only the word itself, but also the POS information and the dependency information of the word helps to extract keyphrases from the text. Similar to the word embeddings, the model uses the POS embedding and dependency embedding lookup table functions to get the POS representation p_i and dependency representation d_i for each word w_i, respectively. Based on these embeddings, we get the word representation $x_i = e_i \oplus p_i \oplus d_i$ for each word w_i.

Then the input representation $\{x_1, x_2, \ldots, x_n\}$ is fed into the Bi-LSTM layer, and the LSTM layer outputs more complex feature representations for the input sequences. The LSTMs perform well at capturing long range information, and bidirectional LSTM is introduced to incorporate the past and future information in the sequence data.

Formally, the left LSTM generates the feature representation $\vec{h}_i$ for each x_i by processing the input representation $\{x_1, x_2, \ldots, x_n\}$ from left to right. In the similar way, the right LSTM generates the feature representation $\overleftarrow{h}_i$ from right to left. Then we obtain the final feature representation h_i through the hidden layer:

$$h_i = tanh(W_1[\vec{h}_i \oplus \overleftarrow{h}_i] + b_1) \tag{1}$$

where $\oplus$ denotes the concatenation between two vectors.

Finally, we get the feature representation $h = \{h_1, h_2, \ldots, h_n\}$ for the sentence s, use the CRF to predict the label sequence y.

The input sentence $s = \{w_1, w_2, \ldots, w_n\}$ corresponds to a matrix $Z \in R^{n \times l}$, where n is the size of the input sequence, and l is the number of labels. Given sentence s, $Z_{i,j}$ denotes the score of the word w_i with the j-th label. The score of an output sequence y is defined by:

$$S(s, y) = \sum_{i=0}^{n} (Z_{i,y_i} + T_{y_{i-1},y_i}) \tag{2}$$

where y_i is the label for the word w_i, and T_{y_{i-1},y_i} represents the transition score from the label y_{i-1} to y_i.

Then the conditional probability of the sequence y is defined by:

$$P(y|s) = \frac{exp(S(s, y))}{\sum_{y' \in Y(s)} exp(S(s, y'))} \tag{3}$$

where $Y(s)$ is the set which contains all the possible label sequences of the sentence s.

Given the sentence s, the decoding process aims to find the label sequence with the highest score in $Y(s)$.

Self-training method

Most keyphrase extraction systems employ supervised machine learning method and achieve promising performance. Since the number of training instances in this task is limited and

annotating more data is expensive, semi-supervised learning methods, which can utilize the annotated data and unannotated data, provide a possible way to improve the performance. To take advantage of the unannotated data, we apply self-training method to the neural model. The details of our method is illustrated in Algorithm 1.

Algorithm 1 Self-training Method

```
Input: Training Set L, Unlabeled Data U, Confidence Set C, Probability
Threshold p
Output: Model Parameters Θ
1 Parameters Θ Initialization
2 for i = 1 ... T do
3   Training the model model_i using training set L
4   Predict the unlabeled data set U using model_i
5   for each instance s in U do
6     if P(y|s)>p then
7       C ← e
8     end if
9   end for
10   U = U − C
11   L = L∪C_i; C_1 is the labelled set using Model_i
12   C = ϕ
13 end for
14 return Model Parameters Θ
```

The basis of the self-training method is the BLSTM-CRF model. The model parameters and the training process in Line 3 are the same as the neural model. In each iteration, we first train the model using the training set, and then select the confidence set from the unlabeled data according to the probability of the instance. The probability is given by the trained model using the Eq 3. The instance with the probability higher than the threshold p is added into the confidence set. The selected confidence set with the label given by the trained model is add to training set, and the new training set will be used in the next iteration.

As described in Algorithm 1, the selection of the confidence set C is the key factor that affect the performance of self-training method, and how to measure the confidence of a tagging sequence is crucial to our method. Despite the conditional probability $P(y|s)$ described in Eq 3, we further investigate the following two metrics to measure the confidence of the tagging sequence y:

$$Confidence_1(y|s) = P(y|s) * N \tag{4}$$

$$Confidence_2(y|s) = \sqrt[N]{P(y|s)} \tag{5}$$

where N is the length of sentence s and $P(y|s)$ is the conditional probability of the sequence y.

Training

The BLSTM-CRF model consists of the neural network layers and the CRF layer, and we will explain the training details of BLSTM-CRF network.

Given the training set $\{x_i, y_i\}_{i=1}^{M}$, the max likelihood training objective is formally by

$$S(\Theta) = \frac{1}{M}\sum_{i=1}^{M} logP(y_i|x_i) + \frac{\lambda}{2}\|\Theta\|^2 \tag{6}$$

where x_i is an input sentence and y_i is the corresponding golden label sequence. $P(y_i|x_i)$ is the probability of the golden label sequence as defied in Eq 3. λ is the regularization parameter and Θ denotes all model parameters of the model.

AdaGrad [50] algorithm is employed to update the parameters of our model.

Experiments

Experimental settings

Data and evaluation. We evaluated scientific keyphrase extraction task on two publicly available datasets: ScienceIE corpus provided by SemEval 2017 Task 10 (https://scienceie.github.io/) and ACL keyphrase corpus (https://nlp.stanford.edu/pubs/FTDDataset_v1.txt). The ScienceIE corpus consists of 500 scientific paragraphs, which come from three different domains. These paragraphs are annotated with keyphrase in three categories: TASK, PROCESS, MATERIAL. The dataset is split into three parts, namely 350 notes for training, 50 notes for development and 100 notes for testing. On the other hand, the ACL keyphrase corpus consists of 462 scientific abstracts, and the human-annotated keyphrases in this corpus contain three categories: DOMAIN, TECHNIQUE and FOCUS. We split this dataset into three parts, namely 370 abstracts for training, 46 abstracts for development and 46 abstracts for testing. Table 1 shows the statistics of the two datasets used in the experiments.

According to the source and domains of the ScienceIE dataset, we crawled the unannotated scientific articles in the same three domains (CS, MC and Ph) from ScienceDirect (https://www.sciencedirect.com/) for self-training method. We randomly select 24,030 abstracts as the unannotated dataset from the website, and this dataset contains 156,459 sentences. For the ACL keyphrase corpus, we use the document collection from the ACL Anthology dataset as the unannotated dataset (https://acl-arc.comp.nus.edu.sg/archives/acl-arc-090501d1/data/txt/) [51]. This dataset contains 7,586 abstracts and 47,718 sentences.

Commonly used precision (P), recall (R) and F1 are used as evaluation metrics.

Pre-processing. Since our neural model takes the word embeddings, POS embeddings and dependency embeddings as the input representation, we use the Stanford CoreNLP library [52] to preprocess the raw text in the corpus, and get the tokenization, POS-tagging and dependency parsing information. Our neural models are implemented based on the LibN3L [53] package.

Post-processing. Many keyphrases, especially their first letters, often appear in the uppercase form in scientific articles, and the uppercase features helps to recognize the keyphrases. To improve the performance of our neural model, we apply a simple heuristic rule to the output of the neural model.

Our neural model outputs the keyphrases in the document, along with their categories and indexes. If one keyphrase contains the capital letter, we label all this phrase occur in the document as the keyphrase.

Hyper-parameter settings. We have tuned the hyper-parameters in our neural models on the development set. The hyper-parameters mainly include two parts. One is the structure

Table 1. Statistics of the datasets.

	Training	Dev	Test
ScienceIE			
Sentences	2403	399	851
Keyphrases	6721	1154	2051
ACL			
Sentences	2159	283	272
Keyphrases	2999	389	392

https://doi.org/10.1371/journal.pone.0232547.t001

Table 2. Hyper-parameter settings.

Type	Hyper-parameter
probability threshold p	0.8
Initial learning rate	0.01
Regularization parameter	10^{-8}
dropout rate	0.4
Dim(emb(word))	300
Dim(emb(POS)), Dim(emb(DEP))	25
Hidden layer size	100

https://doi.org/10.1371/journal.pone.0232547.t002

definition of the neural network, including the size of different embeddings and the size of each hidden layer. The other part includes the hyper-parameters used in the training process.

We use GLOVE word embeddings [42] for our word embeddings initialization, and the dimension of word embeddings is 300. POS embeddings and dependency embeddings are randomly initialized and their dimension is set to 25. These randomly initialized embeddings are fine-tuned in the training process of our model, while the pre-trained word embeddings are not fine-tuned in our experiments.

To alleviate the overfitting problem in the training process, the dropout [54] method is applied to our neural model. When the dropout method is used, the F1 score of our model is improved by 5.9% on ScienceIE corpus and 3.6% on ACL corpus. Table 2 lists the details of these hyper-parameters.

Baselines. The following baselines are used for system comparison in our experiments:

- **BLSTM-CRF**: The bidirectional LSTM combined with CRFs have been successfully used in the sequence labelling task, and achieve promising performance. Alzaidy *et al.* (2019) utilized this model and their results showed that it substantially outperformed some strong baselines and previous methods for keyphrase extraction.
- **BERT**: BERT is designed to pretrain deep bidirectional representations by jointly conditioning on both left and right context [44]. It has achieved state-of-the-art results in various NLP applications, including NER. Our BERT-based keyphrase extraction system is implemented based on an open source project NER-BERT-pytorch (https://github.com/lemonhu/NER-BERT-pytorch). The code of our BERT-based baseline is available at https://github.com/RingoTC/BERT-NER-ScienceIE.

Results

Effects of the heuristic rule. Table 3 shows the performance of BLSTM-CRF model with and without the heuristic rule. BLSTM-CRF-H represents the BLSTM-CRF model using the heuristic rule.

Applying the heuristic rule to the output of our BLSTM-CRF model improves the F1 score from 43.7% to 45.1% on the ScienceIE corpus. It indicates that the heuristic rule helps the

Table 3. Effects of the heuristic rule.

Models	ScienceIE (P/R/F1)	ACL (P/R/F1)
BLSTM-CRF	47.4/40.5/43.7	40.7/31.4/35.4
BLSTM-CRF-H	47.3/43.1/45.1	39.9/31.4/35.1

https://doi.org/10.1371/journal.pone.0232547.t003

Table 4. Effects of the self-training method.

Models	ScienceIE (P/R/F1)	ACL (P/R/F1)
Baseline	47.3/43.1/45.1	40.7/31.4/35.4
$P(y\|s)$	49.2/42.6/45.7	42.6/30.6/35.6
$P(y\|s)^*N$	48.6/43.2/45.7	45.0/29.6/35.7
$\sqrt[N]{P(y\|s)}N$	48.6/42.9/45.6	45.4/31.6/37.3

https://doi.org/10.1371/journal.pone.0232547.t004

model to recognize more keyphrases. Especially, the use of this rule helps to improve the recall of the model. It improves the recall of the neural model from 40.5% to 43.0%, while their precisions are almost the same (47.3% vs 47.4%). This is reasonable since tokens with uppercase letters are often proper nouns or abbreviations for specific terms. These words are often keyphrases in the document.

However, this heuristic rule does not help to improve the performance on the ACL corpus. The likely reason is that the ACL test set contains less keyphrases than the ScienceIE corpus, and there are few cases that satisfy this heuristic rule. Thus, we will not apply the heuristic rule to the BLSTM-CRF model on the ACL corpus in the following experiments.

Self-training method. Furthermore, we introduce self-training method to the neural model to leverage the unannotated data. It selects instances with high confidence from the unlabeled data set, and adds them into the training set.

Despite the conditional probability $P(y|s)$, different confidence functions described in Eqs 4 and 5 are investigated in the experiments. The baseline neural model used for the ScienceIE corpus is the BLSTM-CRF model with the heuristic rule, while the BLSTM-CRF model without the heuristic rule is used for the ACL corpus.

When different confidence functions are used in the self-training method, their F1 scores are very close. According to the results listed in Table 4, we choose the conditional probability $P(y|s)$ as the confidence function for the ScienceIE corpus, and $\sqrt[N]{P(y|s)}$ is used for the ACL corpus.

The increase of the training data helps to extract scientific keyphrases. When the self-training method is applied to the BLSTM-CRF model using heuristic rule, it improves the F1 score from 45.1% to 45.7% on the ScienceIE corpus. Considering the ACL corpus, it improves the F1 score of the BLSTM-CRF model from 35.4% to 37.3%.

Effects of embeddings. Considering the input layer of our neural model, it gets the input word representation by concatenating word embeddings, POS embeddings and dependency embeddings. To study the effects of POS and dependency embedding, we conduct ablation test to show the impact of these embeddings. The baseline model is the BLSTM-CRF model. Table 5 shows the results of the ablation test on the ScienceIE and ACL corpus.

From the Table 5, we can see that both POS embedding and dependency embeddings contribute to the scientific keyphrase extraction. Compared with the model without these embeddings, the system performs slightly better on the ScienceIE corpus. Consider the ACL corpus,

Table 5. Ablation test for different embeddings.

Models	ScienceIE (P/R/F1)	ACL (P/R/F1)
BLSTM-CRF	47.4/40.5/43.7	40.7/31.4/35.4
No Pos embeddings	48.2/39.2/43.2	40.9/29.3/34.2
No dependency embeddings	47.6/39.9/43.4	41.5/28.6/33.8

https://doi.org/10.1371/journal.pone.0232547.t005

Table 6. Results of our model on the ScienceIE corpus, together with other top-performance systems.

Models	F1(%)
Gupta [7]	9.8
Tsai [34]	11.9
AI2 [46]	44
Luan *et al.* (2017) [55]	46.6
BLSTM-CRF	43.7
BERT	35.1
SL-BLSTM-CRF-H	**45.7**

https://doi.org/10.1371/journal.pone.0232547.t006

the BLSTM-CRF model outperforms the model without the POS or dependency embeddings, with the improvement of 1.2% and 1.6%, respectively.

In addition to the word information itself, incorporating POS and dependency information into the BLSTM-CRF model has the potential to improve the performance. For example, BLSTM model can recognize "[main/JJ corrosion/NN products/NNS]" as the Material keyphrase, while it can not extract this keyphrase without its POS information. Its POS information helps to improve the performance of the BLSTM-CRF model.

Comparison with previous systems. Tables 6 and 7 show the results of our model on the ScienceIE and ACL corpus respectively, together with previous state-of-the-art performance systems for comparison.

Gupta [7] and Tsai [34] are unsupervised learning methods. Gupta uses pattern-based bootstrapping approach for automatic keyphrase extraction, and Tsai incorporates hand-designed features into unsupervised bootstrapping framework. AI2 [46] is the best system participating in SemEval 2017 scientific keyphrase extraction task. It employs neural networks and CRFs for this task, and adds scientific terms from external resources as features. Luan *et al.* (2017) [55] has achieved the best keyphrase extraction performance on ScienceIE corpus, and it introduces inductive and transductive semi-supervised learning to the neural tagging models.

From the Table 6, we can see that the performance of the supervised learning and semi-supervised learning methods is much higher than the performance of the unsupervised learning methods on the ScienceIe corpus. The human-annotated data allows the use of supervised learning and semi-supervised learning methods, and helps to improve the scientific keyphrase extraction task.

Among the 17 teams participating the SemEval 2017 scientific keyphrase extraction subtask, we find that the BLSTM-CRF based neural networks achieve top performance in this task. The F1 score of the BLSTM-CRF model used in this paper is slightly lower than that of AI2, which is the best system participating in SemEval 2017 ScienceIE task. The likely reason is that our baseline model performs keyphrase extraction without any external knowledge resources and feature engineering.

Our SL-BLSTM-CRF-H model achieves competitive performance compared with the best performance systems on the ScienceIE corpus. It achieves 2% and 10.6% improvements of F1

Table 7. Results of our model on the ACL corpus, together with other top-performance systems.

Models	F1(%)
BLSTM-CRF	35.4
BERT	31.6
SL-BLSTM-CRF	**37.3**

https://doi.org/10.1371/journal.pone.0232547.t007

score over BLSTM-CRF and BERT, respectively. This demonstrates the effectiveness of our Bi-LSTM-CRF for this task and the importance of character representation. Considering the ACL keyphrase corpus, our best model SL-BLSTM-CRF outperforms these strong baselines, with the improvements of 1.9% and 5.7% over BLSTM-CRF and BERT, respectively.

Conclusions

In this paper, we represented the neural network based model for recognizing and classifying the keyphrases from scientific documents. We employed the state-of-the-art BLSTM-CRF model for scientific keyphrase extraction. Self-training method, which can leverage the unannotated data, was applied to the neural model to improve the performance.

Experiments on the ScienceIE and ACL keyphrase datasets demonstrated the effectiveness of our models. Without any external knowledge resources and manually feature engineering, our BLSTM-CRF model achieved comparable performance. When applying self-training method to the BLSTM-CRF model, it further improved the performance and achieved competitive performance compared with previous state-of-the-art systems.

Acknowledgments

We would like to thank the handling editor and anonymous reviewers for their valuable and insightful comments.

Author Contributions

Conceptualization: Xun Zhu.

Funding acquisition: Donghong Ji.

Methodology: Xun Zhu.

Project administration: Chen Lyu, Donghong Ji.

Resources: Xun Zhu.

Software: Xun Zhu, Chen Lyu, Han Liao, Fei Li.

Supervision: Donghong Ji.

Writing – original draft: Xun Zhu, Chen Lyu.

Writing – review & editing: Chen Lyu, Donghong Ji.

References

1. Turney PD. Learning Algorithms for Keyphrase Extraction. Inf Retr. 2000; 2(4):303–336. https://doi.org/10.1023/A:1009976227802
2. Hulth A. Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 conference on Empirical methods in natural language processing. Association for Computational Linguistics; 2003. p. 216–223.
3. Witten IH, Paynter GW, Frank E, Gutwin C, Nevill-Manning CG. Kea: Practical automated keyphrase extraction. In: Design and Usability of Digital Libraries: Case Studies in the Asia Pacific. IGI Global; 2005. p. 129–152.
4. Kim SN, Medelyan O, Kan MY, Baldwin T. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In: Proceedings of the 5th International Workshop on Semantic Evaluation; 2010. p. 21–26.
5. Berend G. Exploiting extra-textual and linguistic information in keyphrase extraction. Natural Language Engineering. 2016; 22(1):73–95. https://doi.org/10.1017/S1351324914000126

6. 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 the 11th International Workshop on Semantic Evaluation; 2017. p. 546–555.
7. Gupta S, Manning C. Analyzing the dynamics of research by extracting key aspects of scientific papers. In: Proceedings of 5th international joint conference on natural language processing; 2011. p. 1–9.
8. Hasan KS, Ng V. Automatic keyphrase extraction: A survey of the state of the art. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2014. p. 1262–1273.
9. Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Information processing & management. 1988; 24(5):513–523. https://doi.org/10.1016/0306-4573(88)90021-0
10. Liu F, Pennell D, Liu F, Liu Y. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In: Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics. Association for Computational Linguistics; 2009. p. 620–628.
11. El-Beltagy SR, Rafea A. Kp-miner: Participation in semeval-2. In: Proceedings of the 5th international workshop on semantic evaluation; 2010. p. 190–193.
12. Kim SN, Medelyan O, Kan MY, Baldwin T. Automatic keyphrase extraction from scientific articles. Language resources and evaluation. 2013; 47(3):723–742. https://doi.org/10.1007/s10579-012-9210-3
13. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. Journal of machine learning research. 2011; 12(Aug):2493–2537.
14. Ren Y, Zhang Y, Zhang M, Ji D. Context-sensitive twitter sentiment classification using neural network. In: Thirtieth AAAI Conference on Artificial Intelligence; 2016. p. 215–221.
15. Lyu C, Chen B, Ren Y, Ji D. Long short-term memory RNN for biomedical named entity recognition. BMC bioinformatics. 2017; 18(1):462. https://doi.org/10.1186/s12859-017-1868-5
16. Dong X, Chowdhury S, Qian L, Li X, Guan Y, Yang J, et al. Deep learning for named entity recognition on Chinese electronic medical records: Combining deep transfer learning with multitask bi-directional LSTM RNN. PloS one. 2019; 14(5):e0216046. https://doi.org/10.1371/journal.pone.0216046 PMID: 31048840
17. Wan X, Xiao J. Single Document Keyphrase Extraction Using Neighborhood Knowledge. In: AAAI. vol. 8; 2008. p. 855–860.
18. Grineva M, Grinev M, Lizorkin D. Extracting key terms from noisy and multitheme documents. In: Proceedings of the 18th international conference on World wide web. ACM; 2009. p. 661–670.
19. Zhang Q, Wang Y, Gong Y, Huang X. Keyphrase extraction using deep recurrent neural networks on twitter. In: Proceedings of the 2016 conference on empirical methods in natural language processing; 2016. p. 836–845.
20. Yih Wt, Goodman J, Carvalho VR. Finding advertising keywords on web pages. In: Proceedings of the 15th international conference on World Wide Web. ACM; 2006. p. 213–222.
21. Nguyen TD, Kan MY. Keyphrase extraction in scientific publications. In: International conference on Asian digital libraries. Springer; 2007. p. 317–326.
22. Medelyan O, Frank E, Witten IH. Human-competitive tagging using automatic keyphrase extraction. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3. Association for Computational Linguistics; 2009. p. 1318–1327.
23. Zhang Y, Milios E, Zincir-Heywood N. A comparative study on key phrase extraction methods in automatic web site summarization. JDIM. 2007; 5(5):323–332.
24. Hasan KS, Ng V. Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics; 2010. p. 365–373.
25. Wu Z, Giles CL. Measuring term informativeness in context. In: Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies; 2013. p. 259–269.
26. Campos R, Mangaravite V, Pasquali A, Jorge AM, Nunes C, Jatowt A. A text feature based automatic keyword extraction method for single documents. In: European Conference on Information Retrieval. Springer; 2018. p. 684–691.
27. Mihalcea R, Tarau P. Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing; 2004. p. 404–411.
28. Bougouin A, Boudin F, Daille B. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction. In: Proceedings of the Sixth International Joint Conference on Natural Language Processing; 2013. p. 543–551.

29. Boudin F. 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); 2018. p. 667–672.

30. Shi W, Zheng W, Yu JX, Cheng H, Zou L. Keyphrase extraction using knowledge graphs. Data Science and Engineering. 2017; 2(4):275–288. https://doi.org/10.1007/s41019-017-0055-z

31. Yu Y, Ng V. WikiRank: Improving Keyphrase Extraction Based on Background Knowledge. In: International Conference on Language Resources and Evaluation; 2018. p. 3723–3727.

32. Wang R, Liu W, McDonald C. Using word embeddings to enhance keyword identification for scientific publications. In: Australasian Database Conference. Springer; 2015. p. 257–268.

33. Mahata D, Kuriakose J, Shah R, Zimmermann R. 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); 2018. p. 634–639.

34. Tsai CT, Kundu G, Roth D. Concept-based analysis of scientific literature. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management. ACM; 2013. p. 1733–1738.

35. Elman JL. Finding Structure in Time. Cognitive Science. 1990; 14(2):179–211. https://doi.org/10.1207/s15516709cog1402_1

36. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation. 1997; 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

37. Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:150801991. 2015;.

38. Ma X, Hovy E. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2016. p. 1064–1074.

39. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural Architectures for Named Entity Recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2016. p. 260–270.

40. Zhang Y, Yang J. Chinese NER Using Lattice LSTM. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2018. p. 1554–1564.

41. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems; 2013. p. 3111–3119.

42. Pennington J, Socher R, Manning C. Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP); 2014. p. 1532–1543.

43. Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. 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); 2018. p. 2227–2237.

44. Devlin J, Chang MW, Lee K, Toutanova K. 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); 2019. p. 4171–4186.

45. Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov RR, Le QV. Xlnet: Generalized autoregressive pretraining for language understanding. In: Advances in neural information processing systems; 2019. p. 5754–5764.

46. Ammar W, Peters M, Bhagavatula C, Power R. The ai2 system at semeval-2017 task 10 (scienceie): semi-supervised end-to-end entity and relation extraction. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017); 2017. p. 592–596.

47. Tsujimura T, Miwa M, Sasaki Y. TTI-COIN at SemEval-2017 Task 10: Investigating Embeddings for End-to-End Relation Extraction from Scientific Papers. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017); 2017. p. 985–989.

48. Lafferty JD, McCallum A, Pereira FCN. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: Proceedings of ICML 2001; 2001. p. 282–289.

49. Alzaidy R, Caragea C, Giles CL. Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents. In: The world wide web conference; 2019. p. 2551–2557.

50. Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research. 2011; 12(Jul):2121–2159.

51. Bird S, Dale R, Dorr BJ, Gibson B, Joseph MT, Kan MY, et al. The acl anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In: International Conference on Language Resources and Evaluation; 2008. p. 1755–1759.

52. Manning C, Surdeanu M, Bauer J, Finkel J, Bethard S, McClosky D. The Stanford CoreNLP natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations; 2014. p. 55–60.

53. Zhang M, Yang J, Teng Z, Zhang Y. Libn3l: a lightweight package for neural nlp. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016); 2016. p. 225–229.

54. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research. 2014; 15(1):1929–1958.

55. Luan Y, Ostendorf M, Hajishirzi H. Scientific Information Extraction with Semi-supervised Neural Tagging. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing; 2017. p. 2641–2651.