Automatic Keyphrase Extraction from Scientific Chinese Medical Abstracts Based on Character-Level Sequence Labeling

Liangping Ding
dingliangping@mail.las.ac.cn
National Science Library, Chinese Academy of Sciences
Beijing, China
Department of Library Information and Archives Management, University of Chinese Academy of Science
Beijing, China

Zhixiong Zhang*
zhangzhx@mail.las.ac.cn
National Science Library, Chinese Academy of Sciences
Beijing, China
Department of Library Information and Archives Management, University of Chinese Academy of Science
Beijing, China
Wuhan Library, Chinese Academy of Sciences
Wuhan, China

Huan Liu
liuhuan@mail.las.ac.cn
National Science Library, Chinese Academy of Sciences
Beijing, China
Department of Library Information and Archives Management, University of Chinese Academy of Science
Beijing, China

Jie Li
lijie201909@mail.las.ac.cn
National Science Library, Chinese Academy of Sciences
Beijing, China
Department of Library Information and Archives Management, University of Chinese Academy of Science
Beijing, China

Gaihong Yu
yugh@mail.las.ac.cn
National Science Library, Chinese Academy of Sciences
Beijing, China

Abstract

Automatic keyphrase extraction (AKE) is an important task for quickly grasping the main points of the text. In this paper, we regard AKE from Chinese text as a character-level sequence labeling task to avoid segmentation errors of Chinese tokenizer. And we initialize our model with pretrained language model BERT, which is released by Google in 2018. We collect data from Chinese Science Citation Database and construct a large-scale dataset from medical domain, which contains 100,000 abstracts as training set, 6,000 abstracts as development set and 3,094 abstracts as test set. We use unsupervised keyphrase extraction methods including term frequency (TF), TF-IDF, TextRank and supervised machine learning methods including Conditional Random Field (CRF), Bidirectional Long Short Term Memory Network (BiLSTM) and BiLSTM-CRF as baselines. Experiments are designed to compare word-level and character-level sequence labeling approaches on supervised machine learning models and BERT-based models. Compared with character-level BiLSTM-CRF, the best baseline model with F1 score of 50.16%, our character-level sequence labeling model based on BERT obtains F1 score of 59.80%, getting 9.64% absolute improvement. We make our character-level IOB format dataset of automatic keyphrase extraction from scientific Chinese medical abstracts (AKESCMA) publicly available for the benefits of research community, which is available at: https://github.com/possible1402/Dataset-For-Chinese-Medical-Keyphrase-Extraction.

*Corresponding Author

Keywords: Automatic Keyphrase Extraction, Character-Level Sequence Labeling, Pretrained Language Model, Scientific Chinese Medical Abstracts

1 Introduction

Automatic keyphrase extraction (AKE) is a task to extract important and topical phrases from the body of a document [49], which is the basis of information retrieval [27], text summarization [58], text categorization [26], opinion mining [4], and document indexing [16]. It can help us quickly go through large amounts of textual information to find out the main stating point of the text. Appropriate keyphrases can serve as a highly concise summarization of the text and are beneficial to retrieve text.

Classic keyphrase extraction algorithms usually contain two steps [20]. The first step is to generate candidate keyphrases, in which plenty of manually designed heuristics are combined to select potential candidate keyphrases. And the second step is to determine which of these candidate keyphrases are correct.

One of the shared disadvantages in above-mentioned two-step approaches is that the model performance in second step is based on the quality of candidate keyphrases generated in the first step. So some researchers reformulate keyphr-ase extraction as a sequence labeling task and validate the effectiveness of this formulation.

In 2008, Zhang et al. [56] firstly reformulate keyphrase extraction as a sequence labeling task and construct a CRF model to extract keyphrases from Chinese text, which skips the step of candidate keyphrase generation. They use 600 documents to train the model and design lots of features manually. Moreover, they use word-level sequence labeling instead of character-level, tagging the words rather than characters. In Chinese, word is the minimal unit to express semantics. The advantage of word-level formulation is that we can model the relationship among words directly while the disadvantage is that it still depends on the word segmentation results of Chinese tokenizer.

By virtue of automatic extracting features, deep learning methods exceed machine learning methods and gradually become the mainstream in many natural language processing (NLP) tasks. Transformer [50] , an emerging model architecture for handling long-term dependencies, is a substitute to classic neural networks such as Long Short-Term Memory network. In 2018, Google released BERT [13], which is a language model pretrained on large-scale unannotated text and used Transformer to capture deep semantic and syntactic features in text. In 2019, Sahrawat et al.[44] regarded AKE as a sequence labeling task and applied lots of pretrained language models including BERT to English automatic keyphrase extraction task, showing the effectiveness of pretrained language model.

Compared to English keyphrase extraction, Chinese keyphrase extraction is facing with two challenges: lacking of publicly available annotated dataset and relying on Chinese word segmentation tool. Firstly, supervised methods need ground-truth keyphrases of the text to train the model, while there are few Chinese publicly annotated keyphrase extraction datasets, which makes it difficult to do objective evaluation among different researches. Secondly, English tokens is split by white space while there is no delimiter among Chinese words.

To address the above-mentioned challenges, in this paper, we construct a high quality dataset for Chinese automatic keyphrase extraction. We formulate keyphrase extraction from scientific Chinese medical abstracts as a character-level sequence labeling task which doesn't rely on Chinese tokenizer. And also we design experiments to compare the model performance under word-level and character-level sequence labeling formulations, which has not been explored. In addition, for scientific Chinese medical abstracts, English words are interspersed with Chinese words, which increases the difficulty of data preprocessing. So we use Unicode Coding to distinguish English and Chinese, which regards each English word as the elementary unit and each Chinese character as the elementary unit.

Our key contributions are summarized as follows:

1. We regard AKE from scientific Chinese medical abstracts as a character-level sequence labeling task and fine-tune the parameters of BERT[13] to make it adapt to our large-scale keyphrase extraction dataset. Our approach skips the step of candidate keyphrase extraction and is independent of Chinese tokenizer. And also we transfer the pretrained language model BERT to downstream Chinese AKE task without complicated manually-designed features.
2. We design comparative experiments against word-level and character-level sequence labeling formulation for Chinese keyphrase extraction to verify the effectiveness of character-level formulation, especially under the general trend of pretrained language model. The comparative experiments are conducted on machine learning baseline models and BERT-based model. We find that the performance of character-level formulation is comparable to word-level formulation or even higher for traditional machine learning algorithms while has overwhelming advantages for pretrained language model.
3. We process data from Chinese Science Citation Database and construct a large-scale character-level dataset for AKE from scientific Chinese medical abstracts. The

dataset is labeled using Inside–Outside–Beginning tagging scheme (IOB format) [43], which is a common tagging format in chunking tasks such as named entity recognition task. Our proposed dataset contains 100,000 abstracts in training set, 6,000 abstracts in development set and 3,094 abstracts in test set. We make our processed large-scale dataset (AKESCMA) publicly available for the benefits of the research community.

2 Related Work

2.1 Automatic Keyphrase Extraction

Automatic keyphrase extraction has received lots of attention for more than 20 years. Over this time, existing classic methods usually contain two steps: generating candidate keyphrases and determining which of these candidate keyphrases match ground-truth keyphrases.

In the first step, candidate keyphrases generation relies on some heuristics such as extracting n-grams that appears in external knowledge base[18][38], extracting phrases that satisfy pre-defined lexical patterns [2][24][32][52]. The classic approaches in second step can be divided into two categories: unsupervised approaches and supervised approaches.

Unsupervised approaches can be divided into four types: statistics-based approaches[6], graph-based approaches[39][18], embedding-based approaches[35][34] and language model-based approaches[47]. Graph-based methods are the most popular ones while statistics-based methods still hold the attention of the research community.[40]

As for Statistics-based approaches, these approaches don't need any training corpus and they are based on statistical features of the given text such as word frequency [36], TF*IDF [46], PAT-tree [9] and word co-occurrences [37]. And it's suitable for one single document because no prior information is needed. In 1995, Cohen used N-gram statistical information to automatically index the document [10]. It doesn't use any stop list, stemmer or domain-specific external information, allowing for easy application in any language or domain with slight modification. In 1997, Chien used PAT-tree and mutual information between words to extract Chinese keyphrases [9]. In 2009, Carpena et al. considered word frequency and spatial distribution features that keywords are clustered whereas irrelevant words distribute randomly in text [8]. These statistical approaches are usually easy to transfer to a new domain because no prior information is applied.

As for graph-based approaches, keyphrase extraction is a ranking problem substantially. The model scores each candidate for its likelihood of being a ground-truth keyphrase and returns top-ranked keyphrases by setting a threshold. There are lots of popular unsupervised learning algorithms for keyphrases extraction, such as TextRank [39], LexRank [15], TopicRank [5], SGRank [12] and SingleRank [51].

As for supervised approaches, classic keyphrase extraction is formulated as a binary classification problem [16][48] to determine whether the potential candidate keyphrases match ground-truth keyphrases for the text or not. Traditional machine learning algorithms such as Naïve Bayes [54], maximum entropy [61], decision trees [49], SVM [59], bagging [24], boosting [25] rely heavily on complicated manually-designed features which can be broadly divided into two categories: within collection features and external resource-bases features [20]. Within collection features use textual features within training data and can be further divided into statistical features such as term frequency [24], TF*IDF [45], syntactic features such as some linguistic patterns [29] and structural features such as location that keyphrases occur in [52]. External resource-based features consist of lexical knowledge bases such as Wikipedia [18][38], document citations [7], hyperlinks [28]. These methods have some weaknesses. The prediction for each candidate keyphrase is independent to that of others, which means that the model can't capture the connection among keyphrases.

These two-step keyphrase extraction approaches have some drawbacks. Firstly, error propagation. The candidate keyphrases generation errors occurring in the first step will be passed to the second step and influence the performance of the downstream methods. Secondly, the model performance relies heavily on some heuristic settings such as threshold, external resources (Wikipedia, domain ontology, lexicon dictionary etc.), and filtration patterns of POS tags, which make it difficult to transfer to a new domain. Thirdly, it's not able to find an optimal N value (number of keyphrases to extract for the text) based on article contents so it is usually set to a fixed parameter which results in keyphrase extraction performance varying with the value for N. Fourthly, the number of keyphrases is same among text, ignoring the physical truth and bringing lots of redundant keyphrases or losing lots of important keyphrases. Finally, in the second step, the model just analyzes the semantic and syntactic properties of candidate keyphrases separately while losing the meaning of the whole text.

Zhang et al.[56] first reformulates keyphrase extraction to a sequence labeling task, and utilizes user-defined tagging scheme to annotate each word in Chinese text and indicates its chunk belonging. And they use Conditional Random Field model, which shows great performance in sequence labeling task. They design lot of manually-designed features such as POS tagging, TF*IDF, and other location features. Li et al. [60] also uses word-level sequence labeling model to extract keyphrases in automotive field for Chinese text.

Casting keyphrase extraction as a sequence labeling task bypasses the step of candidate keyphrases generation and provides a unified method for automatic keyphrase extraction. Moreover, in sequence labeling, keyphrases are correlated to each other instead of being independent units.

Supervised machine learning methods require precise feature engineering and they rely heavily on manually-designed features, which are time-consuming. Using deep learning method to automatically extract features has become the mainstream of many natural language processing tasks. There are some practices for English AKE. In 2016, Zhang et al. [57] casts keyphrase extraction as a sequence labeling task and proposes a joint-layer recurrent neural network model to extract keyphrases from tweets, which doesn't need complicated feature engineering. In 2019, Sahrawat et al. [44] constructs a BiLSTM-CRF model and uses contextualized word embedding from pretrained language models to initialize the embedding layer. They evaluate model performance on three English benchmark datasets: Inspec [24], SemEval-2010 [30], SemEval-2017 [1] and their model achieves state-of-the-art results on these three benchmark datasets.

Compared with English AKE, Chinese AKE is more complicated owing to the characteristic that there is no delimiter among Chinese words. So there is an additional step in most Chinese AKE models: using Chinese tokenizer to segment words. For traditional two-step keyphrase extraction models, generating Chinese candidate keyphrases needs to use Chinese tokenizer to segment words first. For Chinese AKE models based on sequence labeling, existing methods still use word-level tagging, restricted by the segmentation results of Chinese tokenizer.

2.2 Sequence Labeling Based on BERT

With the improvement of computer hardware and the increase of available data, deep learning based methods gradually occupy the dominant position in the field of natural language processing. Although deep neural networks can learn highly nonlinear features, they are prone to over-fitting without large amount of annotated data. And the objective functions of almost all deep learning architectures are highly non-convex function of the parameters, with the potential for many distinct local minima in the model parameter space[14]. Thus, how to initialize parameters has been a problem that puzzles researchers. The breakthrough comes in 2006 with the algorithms for deep belief networks [21] and stacked auto-encoders[3], which are all based on a similar approach: greedy layer-wise unsupervised pre-training followed by supervised fine-tuning.

Compared with traditional supervised learning tasks that randomly initialize parameters then learn language representations directly from annotated text, pretraining-finetuning mode not only capture the syntactic and semantic features of tokens from large-scale unannotated text but also provide a good initial point for the downstream task, improving the generalization ability of the downstream supervised learning task.

Recently, BERT, short for Bidirectional Encoder Representations from Transformers, which is a pretrained language model receiving widespread concern and is believed to be a milestone in NLP. BERT is pretrained on large-scale unlabeled data from BooksCorpus and English Wikipedia, containing more than 3.3 billion tokens in total. Using BERT to fine-tune the downstream supervised tasks breaks the record for 11 NLP tasks including sentence classification, named entity recognition, natural language inference etc., which proves the feasibility of pretraining-finetuning mode. Using pretrained language models [11][41][42][22][13] has become a standard component of SOTA (state-of-the-art) model architecture in many natural language processing tasks.

Most previous works for sequence labeling are built upon different combinations of LSTM and CRF[17][19][53], Since the release of BERT[13], some researchers show the effectiveness of applying BERT or BERT-based models to sequence labeling task such as named entity recognition task. BERT has a simple architecture based on bidirectional transformers[50], which performs strongly on various tasks depending on its capability to capture long term frequency. Lee et al. introduces BioBERT [33], which is pretrained on large-scale biomedical corpora using the model architecture same with BERT. They test BioBERT on several publicly datasets for named entity recognition such as NCBI disease, BC5CDR. The results show that BioBERT outperforms the state-of-the-art models on six of nine datasets.

In this paper, we combine the benefits of formulating keyphrase extraction from Chinese medical abstracts as a character-level sequence labeling task and the advantage of pretraining-finetuning mode, which can not only avoid errors occurring in Chinese tokenizer, but also extract features automatically rather than using complicated manually-designed features.

3 Methodology

3.1 Task Definition

We cast keyphrase extraction from Chinese medical abstracts as a character-level sequence labeling task and use IOB format as the input format of the model. This task can be formally stated as:

Let $d = \{\omega_1, \omega_2, ..., \omega_n\}$ be an input text, where ω represents the t^{th} element. If the input text is mixed up with Chinese and English, the element is a character for Chinese and a word for English. Assign each ω_t in the text one of the three class labels $Y = \{K_B, K_I, K_O\}$, where K_B denotes that ω_t locates in the beginning of a keyphrase, K_I denotes that ω_t locates in the inside or end of a keyphrase, and K_O denotes that ω_t is not a part of all keyphrases. For example, there is a sentence 'X 连锁先天性肾上腺发育不良患儿的临床及 NR0B1 基因突变分析' and the keyphrases in this sentence are 'X 连锁先天性肾上腺发育不良' and 'NR0B1 基因'.

After IOB format transformation, the character-level tagging result of this sentence is shown in Table 1. As we can see, we split the sentence according to the language which regards each English word as the elementary unit and each

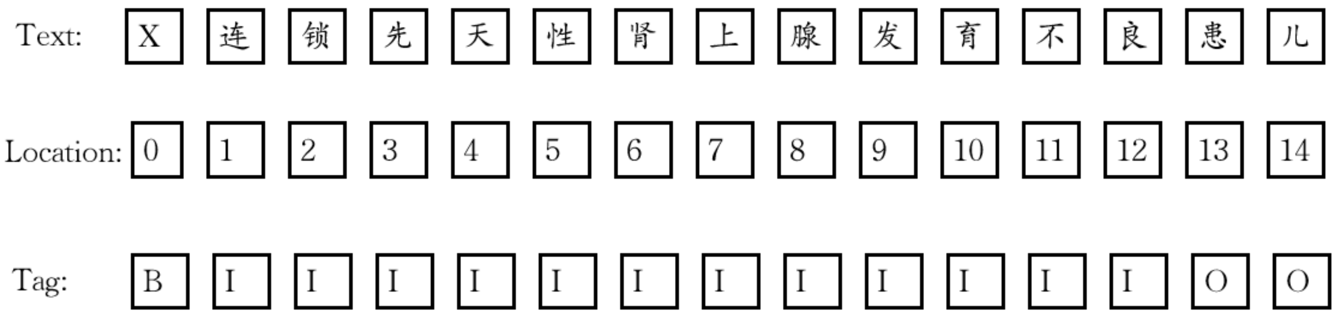

Figure 1. An Example of IOB Format Generation

Table 1. An Example of Character-Level Sequence Labeling

X	连	锁	先	天	性	肾	上	腺	发	育	不	良	患	儿	的	临	床	及	nr0b1	基	因	突	变	分	析
B	I	I	I	I	I	I	I	I	I	I	I	I	O	O	O	O	O	O	B	I	I	O	O	O	O

Chinese character as the elementary unit. This character-level formulation avoids errors of Chinese tokenizer, which has been a troublesome problem in Chinese keyphrase extraction.

3.2 Keyphrase Extraction Evaluation Measures

Although there is a suit of evaluation measures for sequence labeling task, in automatic keyphrase extraction, what we really care about is whether we can extract correct keyphrases of the provided text. So we use precision, recall and F1-score based on actual matching keyphrases against the ground-truth keyphrases for evaluation as used by previous studies [30].

Traditionally, automatic keyphrase extraction system have been accessed using the proportion of top-N candidates that exactly match the ground-truth keyphrases[13]. For keyphrase extraction based on sequence labeling, there is no need for N value and we just use the keyphrases predicted by the model to evaluate the AKE performance. But we need to firstly recognize the keyphrases from IOB format before evaluation. We concatenate characters between label 'B' and the last adjacent label 'I' behind label 'B' as predicted keyphrase.

We denote the total number of predicted keyphrases as r, number of predicted keyphrases matching with ground-truth keyphrases as c, number of ground-truth keyphrases as s. The evaluation measures are defined as follows:

$$Precision : P = \frac{c}{r}$$

$$Recall : R = \frac{c}{s}$$

$$F1 - score : F = \frac{2 \times P \times R}{P + R}$$

3.3 Dataset Construction

We collect data from Chinese Science Citation Database, which is a database contains more than 1000 kinds of excellent journals published in mathematics, physics, chemistry, biology, medicine and health etc. We set some constraints to restrict data to Chinese medical data as well as no incomplete and duplicated records included to ensure the quality of data. The constraints are listed as follows:

1. According to Chinese Library Classification (CLC), the CLC code of medical data starts with the capital letter 'R'. So we restrict data to records that the metadata field of CLC code starts with the capital letter 'R'.
2. The metadata field of language is set to Chinese.
3. The metadata fields of title, abstract and keyphrases are not null. Here, keyphrases refer to author-assigned keyphrases.

Statistics shows that there are 757,277 records meeting the above-mentioned constraints in total. The title and the abstract of each article are concatenated as the source input text. Furthermore, there are two types of keyphrases: extractive keyphrases and abstractive keyphrases. Extractive keyphrases refer to keyphrases that are present in the source input text while abstractive keyphrases refer to keyphrases that are absent in the source input text. Because we formulate keyphrase extraction as a character-level sequence labeling task and can only extract keyphrases that are present in the source input text, we just consider the extractive keyphrases.

For a given text, we expect that all author-assigned keyphrases are extractive keyphrases, so we can annotate as many extractive keyphrases as possible. To achieve that, we firstly match each author-assigned keyphrase with the given text and see if all author-assigned keyphrases can be found in

Table 2. An Example of Word-Level Sequence Labeling

X	连锁	先天性	肾上腺	发育不良	患儿	的	临床	及	nr0b1	基因突变	分析
B	I	I	I	I	O	O	O	O	B	I	O

the text. Then we limit our dataset to records that all author-assigned keyphrases are extractive keyphrases. After filtration, there are 169,094 records in total. We aim to construct a large-scale dataset for our deep neural network model because although deep neural networks can learn highly non-linear features, they are prone to over-fitting compared with traditional machine learning methods.

We choose 100,000 records as our training set, 6,000 records as our development set and 3,094 records as our test set. Training set is used for training the keyphrase extraction model. Development set is used in the training process to monitor the generalization error of the model and to tune hyper-parameters. Test set is used to test the performance of the model. Note that there is no overlap among data sets. Next, we process these three data sets using IOB format to make them suitable for modeling sequence labeling task.

In this paper, we are going to compare word-level and character-level formulation for Chinese keyphrase extraction. So we construct datasets for character-level and word-level sequence labeling separately.

Before generating character-level IOB format for each character, we do some preprocessing steps:

1. Using Unicode Coding to distinguish Chinese and English. To address the problem that English words and Chinese words are mixed together in Chinese medical abstracts, we use Unicode Coding to distinguish English and Chinese. Our proposed data sets can greatly deal with the split of English words and Chinese characters, in which English word and Chinese character is the minimal unit respectively.
2. Converting from all half width to full half width. Punctuations in Chinese medical text include two format: full width and half width. Authors may neglect the format of punctuations, which causes the problem that keyphrases can't match with the abstract. For example, the authors might provide the keyphrase 'er:yag 激光', but they use 'er：yag 激光' in the abstract in which the colon is in full width format. So we transform all half width punctuations to full width punctuations except full stop.
3. Dealing with special characters. There are lots of special characters in scientific Chinese medical abstracts and sometimes there are space characters next to these special characters while sometimes not. To unify the format, we drop all space characters next to special characters.
4. Lowercase. We transform all English words to their lowercase format.

After preprocessing, we do the tagging process, in which we match keyphrases with the source input text to find the locations of keyphrases present in the text and tag the characters within the locations with either label 'B' or label 'I' and characters not within the locations with label 'O'. For the first character in the keyphrase, tag it with label 'B' and for the characters other than the first character in the keyphrase, tag them with label 'I'.

Figure 1 is an example of character-level IOB format generation. In this example, the keyphrase is 'X 连锁先天性肾上腺发育不良'. We match the keyphrase and return the location between 2 and 14. So we tag the character in location 2 with label 'B' and the characters located between 3 and 14 with label 'I'. Other characters not within the location are tagged with label 'O'.

Note that there are two special occasions in our tagging process and we apply some tricks on it.

1. Given two author-assigned keyphrases of the input text, if there is a containment relationship between the location span of two keyphrases, we use Maximum Matching Rule to tag the longest keyphrase. For example:
   Text:' 穴位注射罗哌卡因分娩镇痛对产妇产程的影响'
   This text has two author-assigned keyphrases:' 分娩' and ' 分娩镇痛'. The location span of ' 分娩' is between 8 and 9 while the location span of ' 分娩镇痛' is between 8 and 11. So we tag the characters within the longest keyphrase ' 分娩镇痛' with label 'B' or 'I'.
2. If the first few characters of a keyphrase is equal to the last few characters of the other keyphrase and this keyphrase appears after the other keyphrase in a given text, we will concatenate these two keyphrases by their common characters. For example:
   Text:' 术中经食管超声心动图对心脏瓣膜置换术后即刻人工瓣膜功能异常的诊断价值'
   This text has two author-assigned keyphrases: ' 人工瓣膜' and ' 瓣膜功能异常'. These two keyphrases share common characters ' 瓣膜' and appear next to each other in the text. Then we will tag the keyphrase ' 人工瓣膜功能异常' instead of ' 人工瓣膜' or ' 瓣膜功能异常'. This step determines that our dataset is suitable for flat keyphrase extraction rather than nested keyphrase extraction, which means that each character will be assigned only one label.

For word-level sequence labeling, we use Chinese tokenizer Jieba to segment words. And the tagging process is almost the same with that of character-level dataset construction except that we tag the words rather than characters.

To examine the quality of our data sets, we count the number of recognized keyphrases, the number of correct recognized keyphrases and the number of ground-truth keyphrases in our generated data sets. And we use evaluation measures mentioned in section 3.2 to see the IOB generation performance. The IOB generation results for character-level and word-level are summarized in Table 3 and Table 4 separately.

As we can see, the F1-score of each character-level generated data set is higher than the corresponding word-level generated data set for more than 5 percent. For character-level data sets, owing to the above-mentioned tricks that we apply to IOB generation, the evaluation measures don't reach to 100%. But the character-level IOB generation results on all three data sets still show that our data sets are of good quality. For word-level sequence labeling data sets, the segmentation error of the Chinese tokenizer is a critical reason that the evaluation measures are lower than that of character-level. Take the example mentioned in section 3.1 as an example, the word-level tagging result is shown in Table 2. There is one incorrect keyphrase 'nr0b1 基因突变' which is supposed to be 'nr0b1 基因'. Except for tagged incorrect keyphrases, there might be missing keyphrases because of segmentation error for word-level sequence labeling.

3.4 Model Architecture

We initialize our sequence labeling keyphrase extraction model with pretrained BERT model. The architecture of BERT is based on a multi-layer bidirectional Transformers[50]. Instead of the traditional left-to-right language modeling objective, BERT is pretrained on two tasks: predicting randomly masked tokens and predicting whether two sentences follow each other. Our sequence labeling keyphrase extraction model follows the same architecture as BERT and is optimized on scientific Chinese medical abstracts. We use a feed-forward neural network which acts as a linear classifier layer on top of the representations from the last layer of BERT to compute character level IOB probabilities. Our model architecture is shown in Figure 2.

For a given token, its input representation is constructed by summing the Wordpiece embedding [55], segment embedding and position embedding. The first token of each sequence is always the special token `[CLS]`. The segment embedding is useful in sentence pairs task such as question answering to differentiate sentence. Sentence pairs are separated by a special token `[SEP]` and a sentence `A` embedding is added to each token in the first sentence while a sentence `B` embedding is added to each token in the second sentence. Our task is a single sentence task, so we only use sentence `A` embeddings. The position embedding is used to

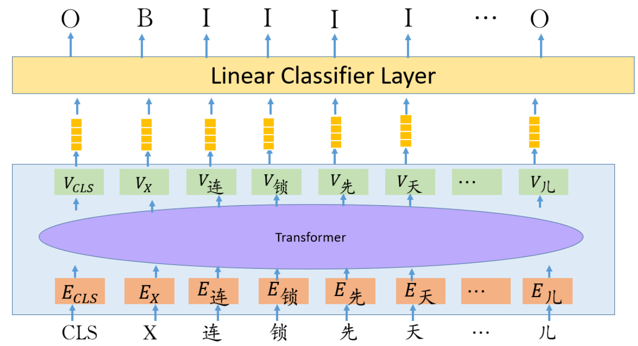

Figure 2. Character-Level Sequence Labeling Keyphrase Extraction Model Architecture

indicate the location of the token in the text and can only take the length lower than 512. A visual representation of our character-level input representations is given in Figure 3.

In addition, BERT can only take the input with the maximum length of 512. Owing to this limitation, some source input text will be truncated, causing the problem that the model might predict some single character as keyphrases. In most cases, single Chinese character makes no sense. We find that some single Chinese characters are meaningful including chemical elements in The Periodic Table such as '氢', 氦', organs such as ' 胃', 脾' and animals such as ' 鼠', 鸡'. So we design a user-defined lexicon to store meaningful Chinese characters for further filtration.

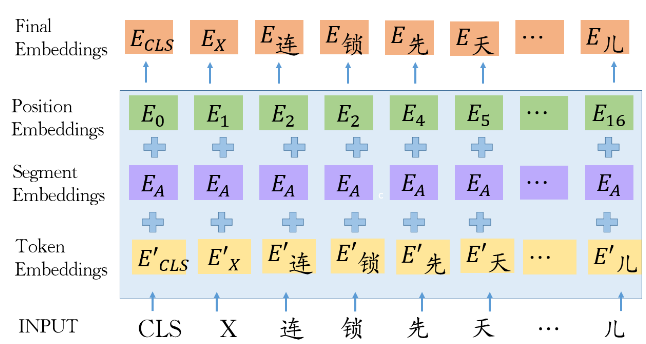

Figure 3. Input Representations of Character-Level Sequence Labeling Keyphrase Extraction Model

4 Experiments & Results

4.1 Experimental Design

In this paper, we firstly conduct unsupervised baseline experiments to demonstrate that traditional unsupervised two-step keyphrase extraction methods are sensitive to N value and the lexicon scale, which depends on precise manual settings. Then before we use sequence labeling formulation to Chinese keyphrase extraction task, we design comparative

Table 3. Character-Level IOB Generation Results on Data Sets

Data Set	P	R	F	number of recognized keyphrases	number of correct recognized keyphrases	number of ground-truth keyphrases
Training Set	99.18%	99.42%	99.30%	416,013	409,371	408,373
Development Set	99.13%	99.54%	99.34%	25,942	26,169	26,061
Test Set	99.15%	99.56%	99.36%	13,344	13,458	13,403

Table 4. Word-Level IOB Generation Results on Data Sets

Data Set	P	R	F	number of recognized keyphrases	number of correct recognized keyphrases	number of ground-truth keyphrases
Training Set	91.15%	96.93%	93.96%	395,852	434,266	408,373
Development Set	91.35%	97.03%	94.11%	25,287	27,680	26,061
Test Set	90.99%	97.11%	93.95%	13,016	14,305	13,403

experiments using word-level and character-level formulation on supervised machine learning baseline methods and BERT-based methods to verify the effectiveness of character-level. Finally, we compare the best unsupervised baseline model, the best character-level machine learning baseline model and our character-level BERT-based sequence labeling keyphrase extraction model to prove the strength of sequence labeling formulation and per-trained language model.

Regarding to unsupervised baselines, We use some traditional approaches including term frequency, TF*IDF based on single document, TF*IDF based on multi-documents, TextRank. Here, TF*IDF based on single document means that we just consider candidate keyphrases' term frequency and inverse document frequency based on one single document. TF*IDF based on multi-documents means that we calculate the statistics based on the whole data set. As we know, the performance of traditional unsupervised approaches varies with the value for N (number of top ranked keyphrases), which is a parameter set manually. And traditional unsupervised Chinese keyphrase extraction relies on Chinese tokenizer to generate candidate keyphrases. Usually, user-defined lexicon will make a great difference to the results of Chinese word segmentation.

So we design two groups of experiments using control variable method for unsupervised baselines according to N value and lexicon scale. Group 1 keeps the same lexicon scale and compares the performance of baseline approaches at different N value of 3 and 5 to ensure the stability of the baseline approaches. Group 2 keeps the same N value and compares the performance of baseline approaches when the lexicon scale for the Chinese tokenizer is different to test the transferability of baseline approaches. We set two kinds of lexicon scales, one using all ground-truth keyphrases in training set, development set and test set as lexicon, the other just using ground-truth keyphrases in training set.

Regarding to supervised machine learning baselines, we cast keyphrase extraction as a sequence labeling task instead of a binary classification task and use CRF, BiLSTM, BiLSTM-CRF algorithms as machine learning baselines.

4.2 Experimental Settings

As for unsupervised baseline approaches, we use Jieba for Chinese word segmentation. Before generating candidate keyphrases, we do some preprocessing steps, such as removing stop words and some special characters. We restrict candidate keyphrases within our user-defined lexicon and noun phrases.

Of the three machine learning baseline approaches, CRF[31] is trained by regularized maximum likelihood estimation and uses Viterbi algorithm to find the optimal sequence of labels. BiLSTM and BiLSTM-CRF[23] are trained with Stochastic Gradient Descent (SGD). The learning rate is set to 5e-4 and the model is trained for 15 epochs with early stopping. The hidden layers are set to 512 units and the embedding size is 768 in both models. In addition, the batch size is set to 64.

For our BERT-based keyphrase extraction model, due to system memory constraints, the batch size is set to 7 and we use SGD to optimize Cross Entropy Loss. The initial learning rate is set to 5e-5 and gradually decreases to 5e-8 as the training progresses and the model is trained for 3 epochs.

In this paper, we use F1-score to evaluate model performance, which is the weighted average of precision and recall, taking both precision and recall into account.

4.3 Unsupervised Baseline Experiments

As for traditional unsupervised baseline experiments, we conduct two groups of baseline approaches comparative experiments according to N value and lexicon scale as what we have mentioned in section 4.1.

Table 5. N-value Comparative Experiments of Unsupervised Baseline Approaches

Method	Top 3 Candidate Keyphrases			Top 5 Candidate Keyphrases		
	P	R	F	P	R	F
Term Frequency	47.66%	33.36%	39.24%	37.53%	43.78%	40.42%
TF*IDF Based on Single Document	50.56%	35.39%	41.61%	38.85%	45.33%	41.84%
TF*IDF Based on Multi Documents	**54.14%**	**37.90%**	**44.59%**	40.37%	47.11%	43.48%
TextRank	43.13%	30.19%	35.52%	33.29%	38.84%	35.85%

Table 6. Lexicon Scale Comparative Experiments of Unsupervised Approaches

Method	P	R	F
Term Frequency(whole lexicon)	47.66%	33.36%	39.24%
Term Frequency(training set lexicon)	37.31%	26.11%	30.72%
TF*IDF Based on Single Document(whole lexicon)	50.56%	35.39%	41.64%
TF*IDF Based on Single Document(training set lexicon)	40.03%	28.03%	32.97%
TF*IDF Based on Multi Documents(whole lexicon)	**54.14%**	**37.90%**	**44.59%**
TF*IDF Based on Multi Documents(training set lexicon)	42.18%	29.53%	34.74%
TextRank(whole lexicon)	43.13%	30.19%	35.52%
TextRank(training set lexicon)	34.37%	24.06%	28.30%

For the group of N value experiments, we restrict the lexicon scale to whole lexicon, which contains author-assigned keyphrases in all the training set, development set and test set as user-defined lexicon for Jieba word segmentation. Table 5 provides the results of N value comparison experiments of baseline approaches. Increasing the N value will improve the recall but lower the precision. We find that the F1-score of baseline approaches varies with the N value, but TF*IDF based on multi-documents achieves best performance among all baseline models no matter the N value. And when the N value is 3, the F1-score of TF*IDF based on multi-documents is 44.59%, which is higher than that when N value is 5.

For the group of lexicon scale experiments, we restrict N value to 3 to compare baseline approaches at different lexicon scales. Table 6 presents the results of lexicon scale comparative experiments of baseline approaches. As we can see, for all unsupervised baseline approaches, the performance of using lexicon that only contains keyphrase in training set for Jieba word segmentation drops at least 7% compared to that of using whole lexicon. The results show that traditional keyphrases extraction approaches for Chinese medical abstracts have poor transferability so when transferring traditional models to a new domain and no lexicon can be used, the keyphrase extraction performance would be poor.

4.4 Word-Level and Character-Level Sequence Labeling Comparative Experiments

We use word-level and character-level sequence labeling dataset separately to train and evaluate supervised machine learning baseline models and BERT-based models.

4.4.1 Supervised Machine Learning Baseline Models.

The F1-score evaluation metrics of word-level and character-level comparative experiments on machine learning baseline models are listed in Table 7. As we can see, word-level sequence labeling formulation is better than character-level sequence labeling formulation for CRF and BiLSTM algorithms while a little bit lower than character-level sequence labeling formulation for BiLSTM-CRF algorithms. The reason might be that BiLSTM-CRF is a more powerful model to capture the contextual relationship among characters to make up for the disadvantage that character-level formulation doesn't model the relationship among words directly.

4.4.2 BERT-based Models.

The precision, recall and F1-score evaluation metrics of word-level and character-level sequence labeling comparative experiments on BERT-based models are listed in Table 8. For word-level sequence labeling formulation, we just use the hidden state corresponding to the first character of the word as input to the linear classifier, which is the same approach used in [13] for named entity recognition task. We find that the precision for word-level is extremely lower than character-level and the F1-score of word-level sequence labeling formulation is more than 20% lower than character-level formulation. Detailed analysis are conducted for this result. We assume that Chinese BERT uses Wordpiece tokenizer which will tokenize each Chinese word into characters in the pretraining process. So Chinese BERT is

Table 7. Word-Level and Character-Level Comparative Experiments of Supervised Machine Learning Baselines

Method	Word-Level	Character-Level
CRF	**47.90%**	46.37%
BiLSTM	**44.35%**	38.38%
BiLSTM-CRF	49.86%	**50.16%**

Table 8. Word-Level and Character-Level Comparative Experiments of BERT-based Models

Metrics	Word-Level	Character-Level
P	26.88%	**60.33%**
R	54.93%	**59.28%**
F	36.10%	**59.80%**

Table 9. Performance Evaluation of Keyphrase Extraction

Method	P	R	F
TF*IDF(Baseline)	54.14%	37.90%	44.59%
BiLSTM-CRF(Baseline)	42.55%	61.09%	50.16%
BERT-based Model(our model)	**60.33%**	**59.28%**	**59.80%**
Adjusted Model(our model)	**61.95%**	**59.22%**	**60.56%**

character-level and has learned good semantic representation of Chinese characters through pretraining, which can maximize the advantages of the character-level sequence labeling formulation and avoid its shortcomings.

4.5 BERT-based Character-Level Experiments

From the results of the above word-level and character-level comparative experiments, we decide to apply character-level formulation to our BERT-based Chinese keyphrase extraction model and the best character-level machine learning baseline model is BiLSTM-CRF. We compare the best unsupervised method TF*IDF with our character-level sequence labeling BiLSTM-CRF model and find that sequence labeling formulation is beneficial for Chinese keyphrase extraction task. And We use character-level BiLSTM-CRF to compare with our character-level BERT-based model. The performance results are summarized in Table 9. Compared with BiLSTM-CRF, our BERT-based model achieves F1-score of 59.80%, exceeding that of baseline approach by 9.64%, which shows that the pretrained language model captures rich features that are useful for downstream keyphrase extraction task. And we remove single Chinese characters that are not in the user-defined lexicon. After removal, the keyphrase extraction performance of our adjusted model reaches to 60.56%.

And we compare the predicted keyphrases with author-assigned ground-truth keyphrases and find that some predicted phrases are concatenation of author-assigned keyphrases. For example, there are two author-assigned keyphrases '卒中' and '抑郁', while our model extracts keyphrases '卒中后抑郁'. Another example, there are two author-assigned keyphrases '急性肠胃炎' and '食源性疾病', while our model extracts keyphrases '食源性胃肠炎'. These examples indicate that as though our model get the F1-score of 59.80%, our model can achieve good practical application performance. In addition, it also indicates that the calculation of evaluation measure is an issue we need to consider further. Using the proportion of predicted phrases that exactly match the ground-truth keyphrases to assess the model is actually not appropriate because there are some biases for author-assigned keyphrases and sometimes the phrases predicted by our model are also concise descriptions for the text.

5 Conclusions

In this paper,we formulate automatic keyphrase extraction as a character-level rather than word-level sequence labeling task and use pretrained language model BERT to fine-tune our keyphrase extraction model on scientific Chinese medical abstracts. Through our experimental work, we prove the benefits of this formulation with this architecture, which bypasses the step of Chinese tokenizer and leverages the power of pretrained language model. In addition, We also design comparative experiments to verify that character-level formulation is more suitable for Chinese keyphrase extraction task under the trend of pretrained language model.

Our approach only deals with keyphrase extraction rather than keyphrase generation, so it can just handle extractive keyphrases. In the future, we plan to build keyphrase generation model to extract keyphrases. And also we will explore the solutions to solve the limitation of BERT's maximum sentence length to avoid being truncated. We expect some of the findings in this paper will provide valuable experiences for automatic keyphrase extraction and other NLP problems like document summarization, term extraction etc.

6 ACKNOWLEDGMENTS

This work is supported by the project "Research on Methods and Technologies of Scientific Researcher Entity Linking and Subject Indexing" (Grant No. G190091) from the National Science Library, Chinese Academy of Sciences and the project "Design and Research on a Next Generation of Open Knowledge Services System and Key Technologies" (2019XM55).

References

[1] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications.

arXiv preprint arXiv:1704.02853 (2017).

[2] Ken Barker and Nadia Cornacchia. 2000. Using noun phrase heads to extract document keyphrases. In *conference of the canadian society for computational studies of intelligence*. Springer, 40–52.

[3] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*. 153–160.

[4] Gábor Berend. 2011. Opinion expression mining by exploiting keyphrase extraction. (2011).

[5] Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction.

[6] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. A text feature based automatic keyword extraction method for single documents. In *European Conference on Information Retrieval*. Springer, 684–691.

[7] Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 2014. Citation-enhanced keyphrase extraction from research papers: A supervised approach. (2014).

[8] Pedro Carpena, Pedro Bernaola-Galván, Michael Hackenberg, AV Coronado, and JL Oliver. 2009. Level statistics of words: Finding keywords in literary texts and symbolic sequences. *Physical Review E* 79, 3 (2009), 035102.

[9] Lee-Feng Chien. 1997. PAT-tree-based keyword extraction for Chinese information retrieval. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*. 50–58.

[10] Jonathan D Cohen. 1995. Highlights: Language-and domain-independent automatic indexing terms for abstracting. *Journal of the American society for information science* 46, 3 (1995), 162–174.

[11] Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*. 3079–3087.

[12] Soheil Danesh, Tamara Sumner, and James H Martin. 2015. Sgrank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. In *Proceedings of the fourth joint conference on lexical and computational semantics*. 117–126.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[14] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research* 11, Feb (2010), 625–660.

[15] Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* 22 (2004), 457–479.

[16] Eibe Frank, Gordon Paynter, Ian Witten, Carl Gutwin, and Craig Nevill-Manning. 1999. Domain-Specific Keyphrase Extraction. (07 1999).

[17] John M Giorgi and Gary D Bader. 2018. Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics* 34, 23 (2018), 4087–4094.

[18] Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In *Proceedings of the 18th international conference on World wide web*. 661–670.

[19] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. 2017. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics* 33, 14 (2017), i37–i48.

[20] Kazi Saidul Hasan and Vincent Ng. 2014. 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)*. 1262–1273.

[21] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation* 18, 7 (2006), 1527–1554.

[22] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* (2018).

[23] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).

[24] Anette Hulth. 2003. 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, 216–223.

[25] Anette Hulth, Jussi Karlgren, Anna Jonsson, Henrik Boström, and Lars Asker. 2001. Automatic keyword extraction using domain knowledge. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 472–482.

[26] Anette Hulth and Beáta B Megyesi. 2006. A study on automatically extracted keywords in text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 537–544.

[27] Steve Jones and Mark S Staveley. 1999. Phrasier: a system for interactive document retrieval using keyphrases. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 160–167.

[28] Daniel Kelleher and Saturnino Luz. 2005. Automatic hypertext keyphrase detection. In *IJCAI*, Vol. 5. 1608–1609.

[29] Su Nam Kim and Min-Yen Kan. 2009. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the workshop on multiword expressions: Identification, interpretation, disambiguation and applications*. Association for Computational Linguistics, 9–16.

[30] 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*. 21–26.

[31] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).

[32] Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu. 2016. Unsupervised keyphrase extraction: Introducing new kinds of words to keyphrases. In *Australasian Joint Conference on Artificial Intelligence*. Springer, 665–671.

[33] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746* (2019).

[34] Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 366–376.

[35] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 257–266.

[36] Hans Peter Luhn. 1957. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development* 1, 4 (1957), 309–317.

[37] Yutaka Matsuo and Mitsuru Ishizuka. 2004. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools* 13, 01 (2004), 157–169.

[38] Olena Medelyan, Eibe Frank, and Ian H Witten. 2009. 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, 1318–1327.

[39] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

[40] Eirini Papagiannopoulou and Grigorios Tsoumakas. 2020. A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10, 2 (2020), e1339.

[41] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

[42] Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. *Technical report, OpenAI* (2018).

[43] Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*. Springer, 157–176.

[44] Dhruva Sahrawat, Debanjan Mahata, Mayank Kulkarni, Haimin Zhang, Rakesh Gosangi, Amanda Stent, Agniv Sharma, Yaman Kumar, Rajiv Ratn Shah, and Roger Zimmermann. 2019. Keyphrase Extraction from Scholarly Articles as Sequence Labeling using Contextualized Embeddings. *arXiv preprint arXiv:1910.08840* (2019).

[45] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.

[46] Gerard Salton, Chung-Shu Yang, and CLEMENT T Yu. 1975. A theory of term importance in automatic text analysis. *Journal of the American society for Information Science* 26, 1 (1975), 33–44.

[47] Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment*. 33–40.

[48] Peter D Tumey. 1999. Learning to extract keyphrases from text. *NRC Technical Report ERB-l 057. National Research Council, Canada* (1999), 1–43.

[49] Peter D Turney. 2000. Learning algorithms for keyphrase extraction. *Information retrieval* 2, 4 (2000), 303–336.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[51] Xiaojun Wan and Jianguo Xiao. 2008. Single Document Keyphrase Extraction Using Neighborhood Knowledge.. In *AAAI*, Vol. 8. 855–860.

[52] Minmei Wang, Bo Zhao, and Yihua Huang. 2016. Ptr: Phrase-based topical ranking for automatic keyphrase extraction in scientific publications. In *International Conference on Neural Information Processing*. Springer, 120–128.

[53] Xuan Wang, Yu Zhang, Xiang Ren, Yuhao Zhang, Marinka Zitnik, Jingbo Shang, Curtis Langlotz, and Jiawei Han. 2019. Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics* 35, 10 (2019), 1745–1752.

[54] Ian H Witten, Gordon W Paynter, Eibe Frank, Carl Gutwin, and Craig G Nevill-Manning. 2005. Kea: Practical automated keyphrase extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*. IGI global, 129–152.

[55] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).

[56] Chengzhi Zhang. 2008. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems* 4, 3 (2008), 1169–1180.

[57] Qi Zhang, Yang Wang, Yeyun Gong, and Xuan-Jing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on twitter. In *Proceedings of the 2016 conference on empirical methods in natural language processing*. 836–845.

[58] Yongzheng Zhang, Nur Zincir-Heywood, and Evangelos Milios. 2004. World wide web site summarization. *Web Intelligence and Agent Systems: An International Journal* 2, 1 (2004), 39–53.

[59] Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. 2011. Topical keyphrase extraction from twitter. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics, 379–388.

[60] 李丽双, 党延忠, 张婧, and 李丹. 2013. 基于条件随机场的汽车领域术语抽取. Ph.D. Dissertation.

[61] 李素建, 王厚峰, 俞士汶, and 辛乘胜. 2004. 关键词自动标引的最大熵模型应用研究. Ph.D. Dissertation.