Design and Implementation of Keyphrase Extraction Engine for Chinese Scientific Literature

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

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

Yang Zhao
zhaoyang@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

Abstract

Accurate keyphrases summarize the main topics, which are important for information retrieval and many other natural language processing tasks. In this paper, we construct a keyphrase extraction engine for Chinese scientific literature to assist researchers in improving the efficiency of scientific research. There are four key technical problems in the process of building the engine: how to select a keyphrase extraction algorithm, how to build a large-scale training set to achieve application-level performance, how to adjust and optimize the model to achieve better application results, and how to be conveniently invoked by researchers. Aiming at the above problems, we propose corresponding solutions. The engine is able to automatically recommend four to five keyphrases for the Chinese scientific abstracts given by the user, and the response speed is generally within 3 seconds. The keyphrase extraction engine for Chinese scientific literature is developed based on advanced deep learning algorithms, large-scale training set, and high-performance computing capacity, which might be an effective tool for researchers and publishers to quickly capture the key stating points of scientific text.

Keywords: Keyphrase Extraction, Artificial Intelligence Engine, Chinese Scientific Literature

ACM Reference Format:
Liangping Ding, Zhixiong Zhang, Huan Liu, and Yang Zhao. 2021. Design and Implementation of Keyphrase Extraction Engine for Chinese Scientific Literature. In *EEKE '21, September 27-30, 2021, Illinois, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Keyphrase extraction task is a branch of information extraction and has been a research hotspot for many years. It aims to identify important topical phrases from text [18], which is of great significance for readers to quickly grasp the main idea of the articles and select the articles that meet their reading interests. Keyphrase extraction task is the basis for many natural language processing tasks such as information retrieval [8], text summarization [22], text classification [7], opinion mining [2], and document indexing [19].

For Chinese scientific literature, there are cases of missing keyphrases stored by publishers. In addition, many keyphrases given by authors do not fully reveal the main idea of the text. So keyphrase extraction for Chinese scientific literature is particularly important, not only to fill the gap of keyphrase metadata fields in publishers' repositories, but also serve as an effective complement to the keyphrases given by authors themselves. It can also provide reference for researchers when writing Chinese scientific papers.

The training corpus used in current Chinese keyphrase extraction models is generally limited to one or several subject areas and is relatively small in size [20], which is

*Corresponding Author

EEKE2021, September 27-30, 2021, Illinois, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

difficult to be oriented to large-scale applications. Moreover, the keyphrase extraction models are generally self-stored by the developers, making it difficult for widespread use by researchers.

To address the above problems, we constructed a keyphrase extraction engine for Chinese scientific literature based on a large-scale training corpus from multiple disciplines for practical applications. The engine can be easily called by means of Application Programming Interface (API) without local model installation and configuration. In this paper, we discuss the overall construction idea of building the keyphrase extraction engine for Chinese scientific literature, the solutions to the key technical problems, and the specific engineering implementation of the engine.

2 Related Work

Currently, the popular keyphrase extraction methods can be divided into three categories: (1) keyphrase extraction based on traditional two-stage ranking; (2) keyphrase extraction based on sequence labeling; (3) keyphrase extraction based on span prediction. The traditional two-stage ranking based methods use some heuristic rules to identify candidate keyphrases from the text in the first stage, and use a ranking algorithm to rank the candidate keyphrases in the second stage. The commonly used ranking algorithms include term frequency [6], TF*IDF [17], etc. A major drawback of this two-stage approach is the error propagation, which means that the error caused in the candidate keyphrases generation will be passed to the candidate keyphrases ranking.

To address this issue, researchers proposed unified keyphrase extraction formulations, which regard keyphrase extraction task as a sequence labeling task or span prediction task. Sequence labeling formulation usually uses BIO [14] or BIOES [15] tagging schemes to annotate tokens in the text sequences, and then train keyphrase extraction models based on machine learning algorithms [20] or deep learning algorithms [21][16]. The idea of span prediction formulation originates from machine reading comprehension based on SQuAD format [13], which predicts the role of tokens in the sequence by training two binary classifiers to determine whether they are the start and end positions of keyphrases [12]. While no consensus has been reached about what kind of formulation should be used for supervised keyphrase extraction task.

In addition, keyphrase extraction algorithm is another important issue that should be paid attention to. In 2018, Google released pretrained language model BERT [3], which attracted widespread attention in the field of natural language processing. This study is widely regarded as a landmark discovery that provides a new paradigm for the field of natural language processing. In the past two years, a large number of pretrained language models have emerged, and many researchers found that using pretrained language models can lead to large improvements in the model performance of downstream tasks [1][9]. Furthermore, some researchers suggested that incorporating external features such as lexicon feature to pretrained language model can further boost the model performance[11][10].

Even though advanced keyphrase extraction algorithms are applied, there are less publicly available keyphrase extraction engine that can be directly called by users to the best of our knowledge, limiting the industrialization of academic achievements. In this paper, we illustrate the construction process of keyphrase extraction engine for Chinese scientific literature, aiming to provide reference for academic researches and industrial usage of keyphrase extraction.

3 The Overall Construction Idea

To build a keyphrase extraction engine for Chinese scientific literature that can be used for practical applications in multiple disciplines, there are four key technical problems: how to select a keyphrase extraction algorithm, how to build a large-scale training set to achieve application-level performance, how to adjust and optimize the model to achieve better application results, and how to be conveniently invoked by researchers.

To address the problem of how to choose an appropriate keyphrase extraction algorithm, we first investigated the current popular and advanced keyphrase extraction algorithms, and used publicly available dataset to compare model performance and determine an optimal keyphrase extraction model for engine construction.

To address the problem of how to construct an application-level large-scale training set, we took advantage of the title, abstract and keyphrase metadata fields of the Chinese Science Citation Database (CSCD) to construct a large-scale training set covering multidisciplinary fields such as medicine and health, industrial technology, agricultural science, mathematical science, chemistry and biological science.

To address the problem of how to adjust and optimize the model to achieve better application results, we used the TF*IDF algorithm as a complement to compensate for the data shortage of humanities domain in the training corpus. And we used large-scale scientific literature as a corpus to calculate the inverse document frequency. Aiming at the problem that keyphrases are often truncated by TF*IDF algorithm, we proposed a circular iterative splicing algorithm to capture more accurate keyphrases.

To address the problem of how to be conveniently invoked by researchers, we deployed the keyphrase extraction model as a service, so that researchers can call the API interface of the model by GET or POST method to obtain the keyphrase extraction results for the given text, without the need for local model installation and configuration.

Table 1. Experimental Results of Keyphrase Extraction Model on CAKE test set

Models	Precision	Recall	F1-score
BERT+SoftMax	63.81%	56.52%	59.94%
BERT+POS+SoftMax	64.83%	57.44%	60.91%
BERT+Lexicon+SoftMax	68.06%	60.67%	**64.15%**
BERT+CRF	64.87%	59.15%	61.88%
BERT+Span	65.51%	57.61%	61.31%

Table 2. Assessment Results of the Training set

Indicators	Results
Precision	99.38%
Recall	97.56%
F1-score	98.46%
Number of correct keyphrases identified	4,447,454
Number of all keyphrases identified	4,447,454
Number of author-given keyphrases	4,558,596

4 Solutions to Key Technical Problems

For the four key technical problems faced in the engine construction process, we proposed the corresponding solutions.

4.1 Selection of Keyphrase Extraction Model

Pretrained language model BERT has captured common language representations from large-scale corpus, enabling downstream supervised learning tasks to achieve great model performance even with a small amount of labeled data. We assumed that taking advantage of pretrained language model, which has been pretrained using large-scale unsupervised text, is of great value to build a keyphrase extraction model for Chinese scientific literature applicable to multi-disciplines. Therefore, we decided to construct a keyphrase extraction model for Chinese scientific literature based on BERT-Base-Chinese, and tried to experiment with both sequence labeling formulation and span prediction formulation to find the optimal keyphrase extraction algorithm for keyphrase extraction engine.

It is worth noting that for Chinese keyphrase extraction, there is no delimiter like space in English to indicate the segmentation of words. So it's necessary to consider whether to use character or word as the minimal language unit to feed into the model. It has been shown that for Chinese keyphrase extraction task, using character as the smallest linguistic unit can achieve better results [4]. In Chinese, word is the smallest unit for expressing semantics. Even though character formulation can avoid the errors caused by Chinese tokenizer, it also loses some of the semantics. To remedy the deficiency, we considered incorporating external features including POS feature and lexicon feature into the model to add in semantics and human knowledge indirectly.

We used the publicly available Chinese keyphrase extraction dataset CAKE [4] for the experiments to determine the best algorithm, which is a dataset containing Chinese medical abstracts from CSCD in sequence labeling format. 100,000 abstracts are included in the training set and 3,094 abstracts are included in the test set. Based on the training set of CAKE, we conducted experiments on five models: $BERT + SoftMax$, $BERT + POS + SoftMax$, $BERT + Lexicon + SoftMax$, $BERT + CRF$, and $BERT + Span$. The first four of these models are based on sequence labeling task formulation, while the last model is based on span prediction formulation. The short description of each model is shown in the following:

1. The $BERT + SoftMax$ model defined the task of keyphrase extraction from Chinese scientific literature as a character-level sequence labeling task, where each token was annotated in BIO tagging scheme. A SoftMax classification layer was added on top of the pretrained language model BERT to output the probability of each category. The parameters of BERT was fine-tuned by CAKE training data.
2. Based on $BERT + SoftMax$ model, we fused POS feature into the embedding space of BERT to incorporate word semantics indirectly and constructed the $BERT + POS + SoftMax$ model. The POS tagging was generated by Hanlp[1]. The details of feature incorporation and model construction are shown in [5].
3. We collected keyphrases from the keyphrase metadata fields in CSCD restricted to medical domain. Based on $BERT + SoftMax$ model, we used BIO tagging scheme to generate lexicon feature and embedded it into BERT to add in domain features and indicate word boundary information to some extent, composing $BERT + Lexicon + SoftMax$ model.
4. The $BERT + CRF$ model used Conditional Random Field (CRF) layer on top of BERT to capture the sequential features among labels. To learn a reasonable transition matrix, we used a hierarchical learning rate, using a learning rate of 5e-5 for training the parameters of the neural network layers of BERT and a learning rate of 0.01 for training the parameters of the CRF layer.
5. $BERT + Span$ model defined keyphrase extraction task as a span prediction problem. Two binary classifiers were trained to determine whether each token is a start position or an end position of the keyphrase.

Table 1 shows the keyphrase extraction performance of the above-mentioned models on the CAKE test set. In the experiments of keyphrase extraction for Chinese scientific literature, we were concerned with how many correct keyphrases we can identify from the given text. Therefore, we compared the keyphrases predicted by the model with the

[1]https://github.com/hankcs/HanLP

Table 3. Statistics of the Discipline Distribution in Training Set

Chinese Library Classification (CLC)	Discipline	Number of Abstracts
R	Medicine and health sciences	421,879
T	Industrial Technology	386,649
S	Agricultural science	142,866
O	Mathematics, physics and chemistry	80,052
Q	Life sciences	60,901
P	Astronomy and geoscience	56,301
X	Environmental science	54,712
F	Economics	27,078
U	Transportation	15,664
V	Aviation and Aerospace	13,956
G	Culture, science, education and sports	7,848
N	Natural science	3,565
C	Social sciences	3,505
B	Philosophy and religions	3,379
K	History and geography	2,001
E	Military science	1,059
D	Politics and law	971
J	Art	712
H	Languages and linguistics	278
Z	General works	48
I	Literature	23
A	Marxism, Leninism, Maoism and Deng Xiaoping theory	12

keyphrases given by the authors and calculated the precision, recall and F1-score to evaluate the model performance. The formula for each indicator is as follows.

$$Precision = \frac{c}{r} \tag{1}$$

$$Recall = \frac{c}{s} \tag{2}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

where c denotes the number of keyphrases predicted by the model that match the author-given keyphrases; r denotes the number of keyphrases predicted by the model in total; and s denotes the number of all author-given keyphrases.

The experimental results showed that the best results can be achieved by adding a SoftMax layer directly on top of the BERT model for classification incorporating the lexicon features simultaneously, which is $BERT+Lexicon+SoftMax$ model. Without adding external features, $BERT + CRF$ model and $BERT + Span$ model achieved better results than $BERT + SoftMax$ model. We finally decided to use the $BERT + Lexicon + SoftMax$ model architecture to build the keyphrase extraction engine for Chinese scientific literature.

4.2 Construction of Application-level Large-scale Training Set

We aimed to build a keyphrase extraction engine for Chinese scientific literature applicable to multidisciplinary fields using large scale training data, while CAKE dataset only contained 100,000 abstracts from medical field, which cannot meet the demand for practical applications. So we constructed a large-scale dataset based on CSCD and evaluated the quality of the dataset. The details of the training set generation are described as follows.

In order to ensure that the constructed training set had a high recall and can annotate as many keyphrases as possible, we processed the tile, abstract and keyphrase fields in the Chinese Science Citation Database and selected the records in which all of the author's given keyphrases appeared in the abstract. Finally, a total of 1,137,945 records were obtained to satisfy the above conditions, and the total number of keyphrases was 1,055,335 (removing duplicates).

We selected 1.1 million records for generating the training set and 37,945 records for generating the test set. Based on the obtained titles, abstracts and keyphrases, we concatenated titles and their corresponding abstracts by period and used BIO tagging scheme to convert the final concatenated text into sequence labeling format, and assigned labels to each token to generate the dataset in the format required for model training. Specifically, given the concatenated text and

keyphrases, we assigned label "B" to the first token of the keyphrase in the text, "I" to the other tokens of the keyphrase, and "O" to the tokens in the text that did not belong to any keyphrase.

To ensure that the training set is of high quality and to avoid providing incorrect supervised signals for model training, we assessed the quality of the training set by comparing author-given keyphrases with the automatic extracted keyphrases in the dataset. The assessment results of the training set are shown in Table 2. It is worth noting that in the process of training set generation, we used the same processing technique as Ding et al. [4] and therefore the quality of the training set cannot reach to 100%. For example, if there was an inclusion relationship between two keyphrases, the longest keyphrase would be selected for labeling; if there was an overlapping relationship between two keyphrases, the two keyphrases would be concatenated according to the overlapping tokens.

In order to ensure that the model can support large-scale applications in multidisciplinary domains, we counted the first-class discipline distribution in the training set based on Chinese Library Classification (CLC), and the statistics are shown in Table 3 [2].

4.3 Model Adjustment and Optimization

Based on the finalized $BERT + Lexicon + SoftMax$ model, we fine-tuned the model using 1.1 million BIO-format Chinese scientific records from multidisciplinary domains. The parameters used in the training process are shown in Table 4[3]. Due to memory limitation, it was not feasible to load the entire dataset into memory, so we transformed the data into the format shown in Figure 1. We loaded the data by Pytorch DataLoader, which read one record at a time by using an iterator, and calculated the gradient of the model after the amount of data reached to the batch size. The final model performance on our all-domain test set are shown in Table 5. It's worth noting that the practical keyphrase extraction results are greater than the statistical indicators because we used exact match principle to calculate the related indicators, while there are some recognized keyphrases not included in the author-given keyphrases but still indicate the main point of the text.

By observing the test results of the model during the practical application, we found that the model did not achieve the expected prediction results for the data in the humanities domain and could not capture the high-frequency words appearing in the text. As shown in Table 3, the training corpus for the humanities domain was small, and apparently the model did not capture enough features on the data from these domains, causing the problem that the number of keyphrases that can be identified for these domains are very limited. To address this issue, we decided to use TF*IDF algorithm as a complement to the extraction results of the $BERT + Lexicon + SoftMax$ model to capture the high frequency keyphrases that appeared in the text.

We randomly selected 1 million abstracts from the Chinese Science Citation Database as the training corpus for the calculation of inverse document frequency (IDF), using Jieba[4] as the Chinese tokenizer to segment the words. To guide the word separation and avoid the professional terms to be cut incorrectly, we introduced all the keyphrases in CSCD, totaling 2,606,322 (no duplicates), as Jieba's user-defined lexicon. The phrases in the custom lexicons as well as nouns in the corpus were calculated for their inverse document frequency, and finally an IDF file was obtained for subsequent keyphrase extraction of Chinese scientific literature based on the TF*IDF algorithm.

Sequence	Label
['meo', '卫', '星', '内', '部', '充', '电', '环', '境', '及', '典', '型', '材', '料', '充', '电', '特', '征', '分', '析', '。'] (Analysis of meo satellite internal charging environment and typical material charging characteristics.)	['B', 'I', 'I', 'B', 'I', 'I', 'I', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']
['阿', '司', '匹', '林', '联', '合', '替', '格', '瑞', '洛', '致', '严', '重', '下', '消', '化', '道', '出', '血', '。'] (Severe lower gastrointestinal bleeding due to aspirin combined with tegretol.)	['B', 'I', 'I', 'I', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'B', 'I', 'I', 'I', 'I', 'I', 'O']

Figure 1. Input Format to DataLoader

Table 4. Parameter Configuration of the Proposed Approach

Parameters	Values
Batch size	7
Epoch	1
Optimizer	Adam
Learning rate scheduler	exponential decay
Initial learning rate	5e-5
Max sequence length	512

Table 5. Keyphrase Extraction Model Performance on All-Domain Test Set

Indicators	Results
Precision	59.11%
Recall	46.84%
F1-score	52.26%
Number of correct keyphrases identified	77,735
Number of all keyphrases identified	131,517
Number of author-given keyphrases	165,956

[2]Some articles have more than one CLC code, the statistics total is over 1.1 million.

[3]Noted that because of computational limitation, the batch size was set to 7 and we assumed that 1 epoch was enough because of the large-scale training set.

[4]https://github.com/fxsjy/jieba

	New Keyphrases	Used Keyphrases	Unused Keyphrases
Iteration 1	① 民用飞机电传飞控系统 (Civilian Aircraft Telemetry Flight Control System) ② 民用飞机适航规范 (Airworthiness Specifications for Civil Aircraft) ③ 电传飞控系统架构设计 (Architecture Design of the Telemetry Flight Control System)	① 民用飞机 (Civilian Aircraft) ② 架构设计 (Architecture Design) ③ 电传飞控系统 (Telex Flight Control System) ④ 适航规范 (Airworthiness Specifications)	① 安全性需求 (Security Requirements) ② 需求论证 (Proof of Need) ③ 具体体现 (Specific Embodiment) ④ 安全要求 (Security requirements)
Iteration 2	None	None	① 民用飞机电传飞控系统 (Civilian Aircraft Telemetry Flight Control System) ② 民用飞机适航规范 (Airworthiness Specifications for Civil Aircraft) ③ 电传飞控系统架构设计 (Architecture Design of the Telemetry Flight Control System) ④ 安全性需求 (Security Requirements) ⑤ 需求论证 (Proof of Need) ⑥ 具体体现 (Specific Embodiment) ⑦ 安全要求 (Security requirements)

Figure 2. Variables in the Iteration Process of the Circular Iterative Splicing Algorithm

At the same time, in order to solve the problem that the keyphrases extracted by TF*IDF algorithm were often truncated, we designed a circular iterative splicing algorithm as improved TF*IDF algorithm. This algorithm spliced two-by-two keyphrases identified by TF*IDF algorithm and determined whether the spliced keyphrases still appeared in the original text. The iterative splicing was continued until no new keyphrases appeared in the original text. We combined the recognized keyphrases of $BERT + Lexicon + SoftMax$ model with that of the improved TF*IDF algorithm as the final keyphrase extraction results for Chinese scientific literature, and the specific process of the model is as follows.

For the given scientific abstract, use $BERT + Lexicon + SoftMax$ model to recognize keyphrases firstly. If the number of the recognized keyphrases of $BERT + Lexicon + SoftMax$ was less than 4, the TF*IDF algorithm would be introduced as a complement. Otherwise, the keyphrase extraction results of the $BERT + Lexicon + SoftMax$ model were returned directly. In the keyphrase extraction process of TF*IDF algorithm, the keyphrases were restricted to nouns or pronouns, etc. to get the top 10 keyphrases in TF*IDF value.

We removed keyphrases which were included in the other keyphrases and the keyphrases whose length were less than two. Then we used the circular iterative splicing algorithm to splice the keyphrases identified by TF*IDF two by two in two directions, splicing from the left and the right. And if the spliced keyphrases still appeared in the text, keep the spliced keyphrases and tag the two original keyphrases as used keyphrases for deletion. Otherwise, keep the keyphrases that are not successfully spliced. This process was iterated until there were no new keyphrases appearing in the original text. The keyphrases identified by the improved TF*IDF algorithm were sorted in descending order according to the TF*IDF value.

The keyphrase extraction results of the $BERT + Lexicon + SoftMax$ model and the improved TF*IDF algorithm were combined and ranked as the final results. The priority of the keyphrases identified by the $BERT + Lexicon + SoftMax$ model were higher than that of the keyphrases identified by the improved TF*IDF algorithm. Based on this principle, we merged the keyphrase extraction results of $BERT + Lexicon + SoftMax$ model with improved TF*IDF algorithm, and took the longest keyphrase for the keyphrases with inclusion relationship. Finally, the top five keyphrases became the final keyphrases. In addition, we used some heuristic rules to filter the final keyphrases, such as removing keyphrases ending with special characters, etc., to improve the accuracy of the keyphrase extraction model.

To further elaborate, for an input abstract [5], the $BERT + Lexicon + SoftMax$ model would process this input first and got the keyphrases as '适航安全性(Airworthiness Safety)', '架构设计(Architecture Design)', '民用飞机(Civilian Aircraft)'. The keyphrases extracted by $BERT + Lexicon + SoftMax$ model were less than four, so the TF*IDF algorithm was trigger to get the top 10 keyphrases according to the TF*IDF value. After the preprocessing, there were eight extracted keyphrases by TF*IDF algorithm as '民用飞机(Civilian Aircraft)', '架构设计(Architecture Design)', '电传飞控系统(Telex Flight Control System)', '安全性需求(Security Requirements)', '适航规范(Airworthiness Specifications)', '需求论证(Proof of Need)', '具体体现(Specific Embodiment)', '安全要求(Security requirements)'. As we

[5] https://kns.cnki.net/kcms/detail/detail.aspx?dbcode=CJFD&dbname=CJFDAUTODAY&filename=HKKX202103004&v=G8TESBUsSe2JeIClg6moqemy3ExscLTVMNxH885u%25mmd2BI%25mmd2Bl9p5i%25mmd2FUmcOUqnMUOyTZM5

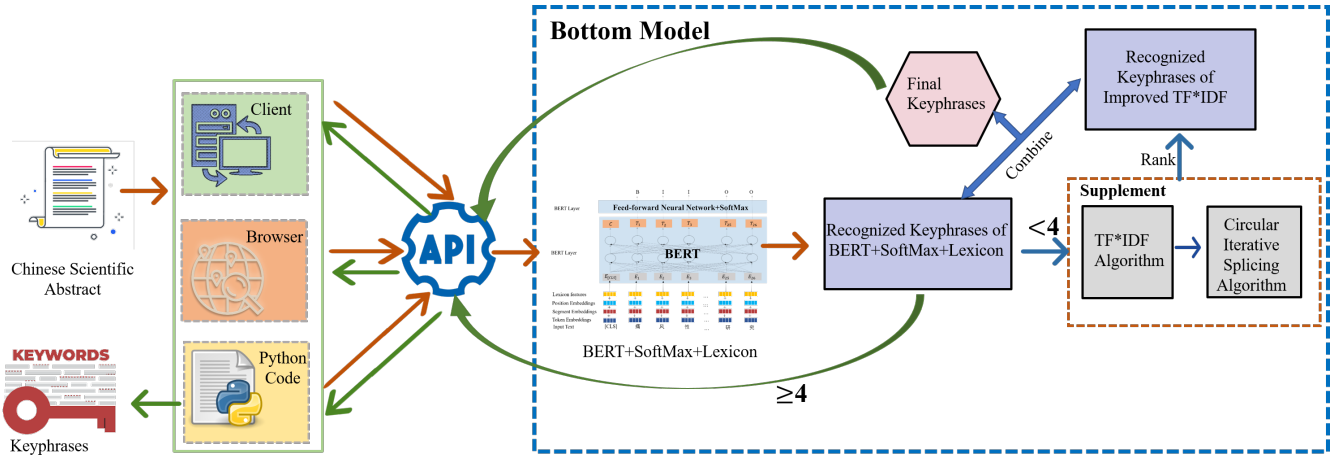

Figure 3. The Processing Flow of the Keyphrase Extraction Engine for Chinese Scientific Literature

can see, there were some redundant keyphrases recognized by traditional TF*IDF, such as '具体体现(Specific Embodiment)'. Next, the circular iterative splicing algorithm would splice the keyphrases two by two. The changes in variables during iteration process are shown in Figure 2.

As we can see, in the first iteration, there were seven spliced keyphrases occurred in the abstract, in which three were new keyphrases and four were original keyphrases (Unused Keyphrases) that didn't splice with other keyphrases. And in the second iteration, no new keyphrases arose, so the iteration finished and all the seven spliced keyphrases kept unused and returned as the results of improved TF*IDF algorithm. Then, we ranked the keyphrases generated by the improved TF*IDF algorithm and combined them with that of *BERT + Lexicon + SoftMax* model to get the further results as '适航安全性(Airworthiness Safety)', '架构设计(Architecture Design)', '民用飞机(Civilian Aircraft)','民用飞机电传飞控系统(Civilian Aircraft Telemetry Flight Control System)', '电传飞控系统架构设计(Architecture Design of the Telemetry Flight Control System)', '安全性需求(Security Requirements)', '民用飞机适航规范(Airworthiness Specifications for Civil Aircraft)', '需求论证(Proof of Need)', '具体体现(Specific Embodiment)', '安全要求(Security requirements)'. Finally, we removed the short keyphrases who had an inclusion relationship with others and got the ultimate top 5 recognized keyphrases as '适航安全性(Airworthiness Safety)', '民用飞机电传飞控系统(Civilian Aircraft Telemetry Flight Control System)', '电传飞控系统架构设计(Architecture Design of the Telemetry Flight Control System)', '安全性需求(Security Requirements)', '民用飞机适航规范(Airworthiness Specifications for Civil Aircraft)'. It can be seen that the final keyphrase extraction results of our proposed hybrid model are better than that of the *BERT + Lexicon + SoftMax* model and the TF*IDF model.

4.4 API Interface Design

In order to avoid various hardware and software constraints that may be encountered in the local deployment of the model, and to provide a fast and convenient way for researchers to invoke the keyphrase extraction model, we deployed the keyphrase extraction model as a service, and built a keyphrase extraction engine for Chinese scientific literature through API interface calls. Researchers can call the API interface of the engine in two ways, POST and GET, to achieve automatic keyphrase extraction of Chinese scientific literature. Pass in the abstract of Chinese scientific literature and the verification code, and the engine would return the keyphrase extraction results in JSON format.

For the GET method, users can send an request to the URL: http://sciengine.las.ac.cn/keywords_extraction_cn to call the keyphrase extraction engine, passing in the abstract of a Chinese scientific literature abstract and the verification code. When the engine receives the call, it will respond by returning the keyphrase extraction results in JSON format. Details of the GET API call are shown in Table VI.

For the POST method, users can send an request to the URL: http://sciengine.las.ac.cn/keywords_extraction_cn to call the keyphrase extraction engine, depositing the abstracts of multiple Chinese scientific articles into a list, and pass in the verification code. After the engine responds, it will return the keyphrase extraction results of all the abstracts in the list in JSON format to achieve batch processing. the details of the POST API call are shown in Table 7.

5 Engineering Implementation

In order to display the keyphrase extraction results intuitively and meet the demands for different users to call the engine, we currently provide three ways to call the keyphrase extraction API interface: browser online demo, Python code

Table 6. GET API Call Details

	Format	Example
Request URL	/keywords_extraction_cn	http://sciengine.las.ac.cn/keywords_extraction_cn
Request Parameters	"data": abstract of Chinese scientific literature, "token": Verification Code	{"data": "新辅助治疗背景下胰腺癌扩大切除术的应用价值。胰腺癌恶性程度高,预后较差,治疗效果仍不理想...(The value of extended resection of pancreatic cancer in the context of neoadjuvant therapy. Pancreatic cancer is highly malignant, with poor prognosis and still unsatisfactory treatment results...)", "token":99999}
Browser parameter	/Keywords_BIO_Lexi?data= &token=	http://sciengine.las.ac.cn/keywords_extraction_cn?data=新辅助治疗背景下胰腺癌扩大切除术的应用价值。胰腺癌恶性程度高,预后较差,治疗效果仍不理想...(The value of extended resection of pancreatic cancer in the context of neoadjuvant therapy. Pancreatic cancer is highly malignant, with poor prognosis and still unsatisfactory treatment results....)&token=99999
Success message	"keywords": [keyphrases list]	{"keywords":["胰腺癌(pancreatic cancer)", "新辅助治疗(neoadjuvant therapy)", "扩大切除术(extended resection)"] }
Error message	"info":error message	{"info": "Server not available!"}, {"info": "Token incorrect!"}

Table 7. POST API Call Details

	Format	Example
Request URL	/keywords_extraction_cn	http://sciengine.las.ac.cn/keywords_extraction_cn
Request Parameters	"data": abstract of Chinese scientific literature, "token": Verification Code	{"data": ["新辅助治疗背景下胰腺癌扩大切除术的应用价值。胰腺癌恶性程度高,预后较差,治疗效果仍不理想...(The value of extended resection of pancreatic cancer in the context of neoadjuvant therapy. Pancreatic cancer is highly malignant, with poor prognosis and still unsatisfactory treatment results...)", "参芪地黄汤联合ACEI/ARB类药物治疗糖尿病肾病的Meta分析...(Meta-analysis of Shenqi Dihuang Decoction combined with ACEI/ARB drugs in the treatment of diabetic nephropathy...)"], "token":99999}
Success message	Abstract ID:[keyphrases list]	{0: ["胰腺癌(pancreatic cancer)", "新辅助治疗(neoadjuvant therapy)", "扩大切除术(extended resection)"], 1: ["糖尿病肾病(diabetic nephropathy)", "ACEI/ARB类(ACEI/ARB)", "META分析(Meta-analysis)", "参芪地黄汤(Shenqi Dihuang Decoction)"]}
Error message	"info":error message	{"info": "Server not available!"}, {"info": "Token incorrect!"}

access and client access. The calling flow of the keyphrase extraction engine for Chinese scientific literature is shown in Figure 3.

5.1 Browser Online Demo

Figure 4. Browser Online Demo Interface

Users can visit the URL: http://sciengine.las.ac.cn/Keywords_BIO_Lexi to test the keyphrase extraction engine online. Type the abstract of a Chinese scientific literature in the input box (it is recommended to use the title + '。' + abstract as input), click the keyphrase extraction button, and the engine API interface will be called automatically to invoke the bottom model and four to five keyphrases related to the main idea will be returned. The response time of the engine is generally within 3 seconds and the interface of the browser online demo is shown in Figure 4.

5.2 Python Code Access

Technical staff who are familiar with Python programming language can download the corresponding sample codes from the website http://sciengine.las.ac.cn/Scripts and revise the file path to achieve convenient usage. There are four files: *keyphrases_extraction_cn_get.py*, the sample code for calling the API interface of the keyphrase extraction engine using GET method; *keyphrases_extraction_cn_post.py*, the sample code for calling the API interface of the keyphrase extraction engine using POST method; *input_cn.txt*, the sample input file; *ReadMe.txt*, the description file.

When using the GET method to call the API interface, input the verification code and the Chinese abstract that needs to be recognized in the corresponding location of the code. Then run *keyphrases_extraction_cn_get.py* file and the automatic keyphrase extraction results will be printed directly. When using POST method to call the API interface, open the *keyphrases_extraction_cn_post.py* file with the Python editor and input the verification code to the corresponding location in the code. Set the paths of the input file and output file, where the format of the input file is one line per abstract. Run *keyphrases_extraction_cn_post.py* file, and the program will read the input file and write the keyphrase extraction results to the output file.

5.3 Client Access

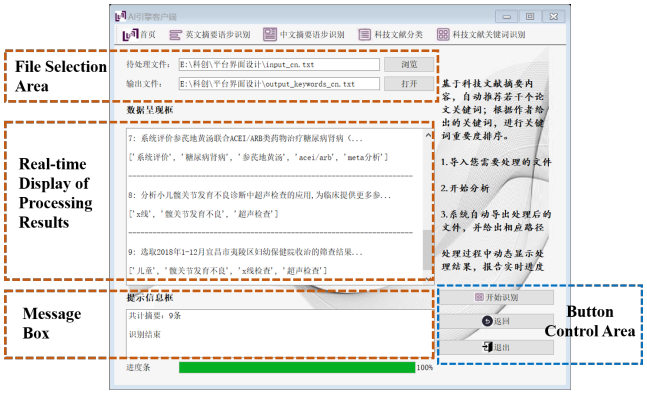

Figure 5. Client Interface

In order to provide for non-technical personnel to use, we designed the client to realize the keyphrase extraction service for Chinese scientific literature without writing a single line of code. Users can download and install the client from the website http://sciengine.las.ac.cn/Client, and get the verification code as the login credentials to call the keyphrase extraction engine API interface to achieve automatic keyphrase extraction of Chinese scientific literature. The keyphrase extraction engine client interface is shown in Figure 5 and the specific operation process is as follows.

1. After opening the client and entering the verification code, click the button of "Keyphrase Extraction for Chinese Scientific Literature" in the menu bar to enter the interface of keyphrase extraction function. Click "Browse" button to import the file to be processed, and it means successful if the data presentation box shows the imported data, and the message box shows the total number of the data.
2. Click "Start Extraction" button, the client will automatically carry out the function of keyphrase extraction for Chinese scientific literature and display the real-time processing progress.
3. When the extraction is finished, the client will pop up completion window and automatically show the output file path.
4. Click "Open" button to view the output file.

6 Conclusions

In this paper, we make full use of the large-scale training corpus of Chinese Science Citation Database and the pretrained language model BERT to construct a keyphrase extraction engine for Chinese scientific literature. We incorporate lexicon features into the high-dimensional vector space of BERT,

fusing human knowledge to instruct the model training. To support practical applications in multidisciplinary fields, the TF*IDF algorithm is introduced as a complement to better capture the high-frequency words appearing in the text. We deploy the engine as a service, which can be invoked using the API interface, and the response speed is generally within 3 seconds. And we provide example scripts in Python for technical staff and a visualization client for non-technical personnel to use without writing a line of code. We hope that our keyphrase extraction engine can provide a feasible path for researchers to improve efficiency.

7 ACKNOWLEDGMENTS

The work is supported by the project “Artificial Intelligence (AI) Engine Construction Based on Scientific Literature Knowledge" (Grant No.E0290906) and the project “Key Technology Optimization Integration and System Development of Next Generation Open Knowledge Service Platform" (Grant No.2021XM45).

References

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676* (2019).

[2] Gábor Berend. 2011. Opinion expression mining by exploiting keyphrase extraction. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, 1162–1170.

[3] 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).

[4] Liangping Ding, Zhixiong Zhang, Huan Liu, Jie Li, and Gaihong Yu. 2020. Automatic Keyphrase Extraction from Scientific Chinese Medical Abstracts Based on Character-Level Sequence Labeling. *Journal of Data and Information Science,* 6, 3 (2020), 33–57.

[5] Liangping Ding, Zhixiong Zhang, and Yang Zhao. 2021. Bert-based Chinese Medical Keyphrase Extraction Model Enhanced with External Features. *International Conference on Asia-Pacific Digital Libraries* (2021).

[6] Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*. 216–223.

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

[8] 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.

[9] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.

[10] Xiangyang Li, Huan Zhang, and Xiao-Hua Zhou. 2020. Chinese clinical named entity recognition with variant neural structures based on BERT methods. *Journal of biomedical informatics* 107 (2020), 103422.

[11] Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 5301–5307.

[12] Funan Mu, Zhenting Yu, LiFeng Wang, Yequan Wang, Qingyu Yin, Yibo Sun, Liqun Liu, Teng Ma, Jing Tang, and Xing Zhou. 2020. Keyphrase Extraction with Span-based Feature Representations. *arXiv preprint arXiv:2002.05407* (2020).

[13] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).

[14] 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.

[15] Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*. 147–155.

[16] 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).

[17] 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.

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

[19] Yi-fang Brook Wu, Quanzhi Li, Razvan Stefan Bot, and Xin Chen. 2005. Domain-specific keyphrase extraction. In *Proceedings of the 14th ACM international conference on Information and knowledge management*. 283–284.

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

[21] 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.

[22] 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.