Deep Web Data Source Classification Based on Text Feature Extension and Extraction

Yuancheng Li, Guixian Wu, and Xiaohan Wang

***Abstract*—With the growth of volume of high quality information in the Deep Web, as the key to utilize this information, Deep Web data source classification becomes one topic with great research value. In this paper, we propose a Deep Web data source classification method based on text feature extension and extraction. Firstly, because the data source contains less text, some data sources even contain less than 10 words. In order to classify the data source based on the text content, the original text must be extended. In text feature extension stage, we use the N-gram model to select extension words. Secondly, we proposed a feature extraction and classification method based on Attention-based Bi-LSTM. By combining LSTM and Attention mechanism, we can obtain contextual semantic representation and focus on words that are closer to the theme of the text, so that more accurate text vector representation can be obtained. In order to evaluate the performance of our classification model, some experiments are executed on the UIUC TEL-8 dataset. The experimental result shows that Deep Web data source classification method based on text feature extension and extraction has certain promotion in performance than some existing methods.**

***Index Terms*—Deep Web, Classification, Attention mechanism, Feature extension.**

I. INTRODUCTION

OVER the past decade, the number of web pages has grown exponentially with the popularity of the Internet [1]. At present, Surface Web refers to resources that can be accessed through static hyperlinks, usually static HTML pages [2]. Such resources can be crawled by web crawlers and are also visible to search engines. Whereas Deep Web refers to resources that are not hidden in the Web database and cannot be crawled by the web crawler. These resources are invisible to the search engine, users who want to get data in it must fill out the form and submit it according to actual needs to dynamically obtain Deep Web resources [3]. Fig. 1 shows an example of the Deep Web. According to statistic, Deep Web has the following advantages compared to Surface Web [4]-[6] (1) Information in Deep Web is 700 to 800 times that of Surface Web information. It includes a large amount of information that traditional search engines cannot find, and its growth rate is much higher than Surface Web; (2) The information contains in Deep Web is of higher quality than the information contained in the Surface Web. Moreover, Deep Web contains information in all areas. In the field of integration, structured data has a higher value, and the Deep Web contains information that is typically structured data. (3) Everyone has access to more than 90% of the Deep Web information, and we can get it for free, which greatly facilitates the interconnection of information. Therefore, research on Deep Web information acquisition has higher practical significance and practical value. To make better use of the information in the Deep Web, it is necessary to classify data sources based on content [7]-[8].

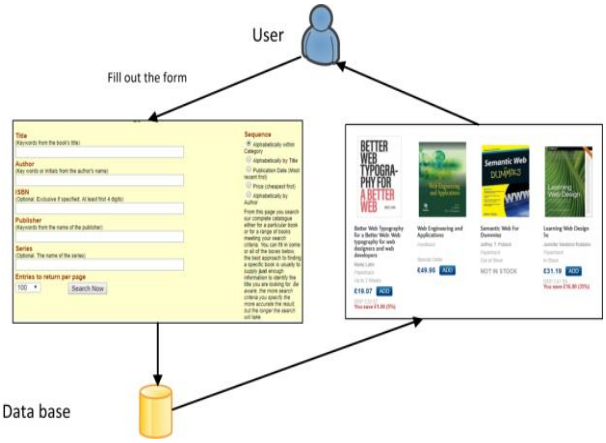


Fig. 1. An example of Deep Web data source.

In recent years, scholars all over the world have propose many kinds of intelligent methods for the classification of data sources. Reference [9] combines the two methods to get the similarity of the search interface and implement classification. The first one is based on vector space. Classic TF-IDF statistics are used to obtain similarities between search interfaces. The other is to use HowNet to calculate the semantic similarity between two pages. Reference [10] proposes a " one hot encoding" method to classify news headlines and summary information collected on the Deep Web. A content-based classification model is proposed in [11], which uses machine learning to filter unwanted information. Word2Vec word embedding tool is used to establish the classification model and classify the selected data set. Reference [12] proposes a new probabilistic subject model to realize text extension and enrich feature description. The deep architecture of the LSTM is applied to Web service recommendations and predictions for more accurate service recommendations. A text categorization network model based on human conditioned reflex (BLSTM) is proposed in [13]. The receptor obtains context information through BLSTM, the nervous center obtains important information of sentences through attention mechanism, and the effector obtains more key information through CNN.

Yuancheng Li , Guixian Wu and Xiaohan Wang are with School of Control and Computer Engineering, North China Electric Power University, Beijing, China. (e-mail: yuancheng@ncepu.cn).

DOI: 10.36244/ICJ.2019.3.7

Reference [14] proposes a coordinated CNN-LSTM-Attention model(CCLA). The semantic and emotional information of sentences and their relationships are adaptively encoded into vector representations of documents. Softmax regression classification is used to determine the emotional tendency in the text. For short text feature extension, there are two main methods at present [15]-[16]: (1) Using topic models such as potential Dirichlet allocation (LDA), Latent Semantic Analysis (LSA), and pLSA. (2) Using search engines and external knowledge bases like WordNet, HowNet, and Wikipedia.

This paper propose a Deep Web data source classification method based on text feature expansion and extraction. In the feature extension stage, We choose extension words through the N-gram model which is easy to train and does not require an external corpus when during feature extension. Then, in the classification stage, we propose a classification method based on Attention-based Bi-LSTM. LSTM is an improvement on traditional RNN. Based on the RNN model, LSTM adds a cell control mechanism to solve the long-term dependence problem and the gradient explosion cause by excessive sequence length [17]-[18]. However, The LSTM model can only utilize the preceding part of the text and does not use the information below, so some semantic information will be lost. To solve this problem, we replace LSTM with Bi-LSTM, which can use both the above and below information simultaneously. Moreover, it is clear that each word in the text contributes differently to the characteristic representation of the text, so whether using the average output of each neuron in the network output layer or the output value of the last neuron, the vector representation of the text cannot be accurately obtained. Therefore, the best way is to use a weighted average to process the output of each neuron in the output layer. To achieve a weighted average, we use the Attention mechanism to handle the output of the Bi-LSTM network. In summary, we propose the deep Web data source classification model based on N-gram and Attention-based Bi-LSTM. Then we conducte multiple sets of comparative experiments on the UIUC TEL-8 dataset. The experimental results show that the Deep Web data source classification method based on n-gram and Attention-based Bi-LSTM has better performance than existing methods.

II. MATERIALS AND METHODS

A. N-gram language model

The N-gram language model plays a pivotal role in natural language processing. Especially in many NLP tasks such as machine translation, syntactic analysis, phrase recognition, part-of-speech tagging, handwriting recognition, and spelling correction.

For a sentence S consisting of n words, the probability of its appearance is:

$$\begin{aligned} \mathrm{P(S)} &= P(w_1, w_2, \cdots, w_n) \\ &= \mathrm{P}(w_1) * \mathrm{P}(w_2|w_1) * \cdots * \mathrm{P}(w_n|w_1 w_2, \cdots, w_{n-1}) \\ &= \prod_{i=1}^{n} \mathrm{P}(w_i|w_1 w_2, \cdots, w_{i-1}) \end{aligned} \quad (1)$$

The probability that (1) represents the ith word is determined by the previous i-1 words. However, a serious problem with this calculation method is that as the length of the sentence increases, the number of parameters that need to be trained will increase exponentially. To solve this problem, according to the Markov hypothesis, supposing that the appearance of the ith word is only related to the first n-1 words. Then, the probability of the sentence $\mathrm{S} = \mathrm{w_1 w_2 \cdots w_n}$ is:

$$\begin{aligned} \mathrm{P(S)} &= P(w_1, w_2, \cdots, w_n) \\ &= \mathrm{P}(w_1) * \mathrm{P}(w_2|w_1) * \cdots * \mathrm{P}(w_n|w_1 w_2, \cdots, w_{n-1}) \\ &= \prod_{i=1}^{n} \mathrm{P}(w_i|w_1 w_2, \cdots, w_{i-1}) \\ &\approx \prod_{i=1}^{n} \mathrm{P}(w_i|w_{i-N+1} w_{i-N+2}, \cdots, w_{i-1}) \end{aligned} \quad (2)$$

The above is the N-gram model. When N = 2, it is assumed that the appearance of each word is only related to one of the previous words, called Bi-gram, as shown in (2).

$$P(S) \approx \prod_{i=1}^{n} \mathrm{P}(w_i|w_{i-1}) \quad (3)$$

In (3),

$$\mathrm{P}(w_i|w_{i-1}) = \frac{c(w_{i-1} w_i)}{c(w_{i-1})} \quad (4)$$

Where $\mathrm{c(w_{i-1} w_i)}$ refers to the number of occurrences of the word sequence $\mathrm{w_{i-1} w_i}$ in the training set, and $\mathrm{c(w_{i-1})}$ refers to the number of occurrences of the word $\mathrm{w_{i-1}}$.

The performance of the models is different when choosing different N. The Bi-gram model is widely used in NLP. The larger N is, the more constraints appear on the next word, and the stronger the recognition ability of the language model, but the higher the complexity of model training, the more sparse the parameters. Conversely, if N is smaller, the language model is easier to train, and the parameters obtain from the corpus will be more, and the statistical information of the corpus can be better utilized. In the research and practical application of natural language processing, the Bi-gram model is the most used.

B. Bidirectional Long Short-Term Memory Network

Long short-term memory (LSTM) is an improvement of the recurrent neural network (RNN), which effectively solves the problem of disappearing gradients. LSTM solves the disappearing gradient problem in RNN by adding a gating function to the general recursive neural network [19]-[20]. Fig. 2 shows the structure of the LSTM cell.

As shown in Fig. 2, The LSTM cell is mainly composed of three parts: the input gate, the forgotten gate, and the output gate. Each gate consists of a sigmoid layer and a vector operation. The probability value of the sigmoid layer output is between 0 and 1, which describes how much each part can pass. The input gate allows the input signal to change the state of the memory unit or block it. Besides, the output gate allows the state of the memory cell to affect other neurons or block it. Finally, the forgotten gate allows the unit to remember or forget its previous state [21].

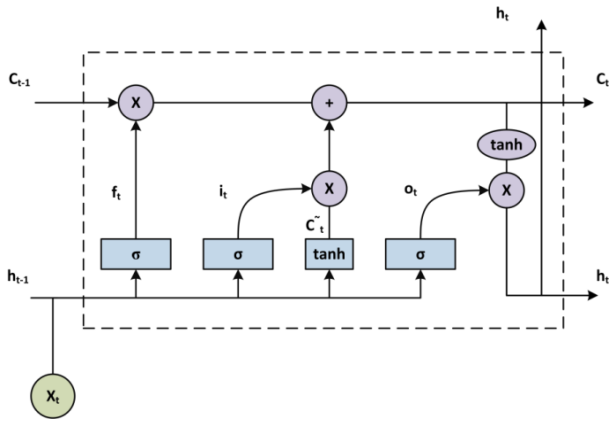


Fig. 2. The structure of a LSTM cell.

The calculation process of a LSTM memory cell is described as follow:

Let vector sequence $X = \{x_1, x_2, \ldots, x_t, \ldots, x_n\}$ is the input of LSTM network, vector sequence $H = \{h_1, h_2, \ldots, h_t, \ldots, h_n\}$ is the output of hidden layer of LSTM network and C_i, $i = 0,1, \ldots, t, \ldots n$ is the state of the i-th memory cell.

1. Decide how much to forget from the state of last memory cell, the calculation process is as follows.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t]_f + b_f\right) \tag{5}$$

In (5), σ is sigmoid function, W_f and b_f are respectively weights and bias of forget gate, $f_t \in (0,1)$.

2. Decide what to add to the cell state, the calculation process is as follows.

$$i_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_i\right) \tag{6}$$

$$\tilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{7}$$

In this step $\tilde{C}_t$ is the update candidate who calculates by output of last cell and input of current cell, and $i_t \in (0,1)$ decide which part of the candidate value is added to the state of the cell.

3. Update cell state

Update C_{t-1} to C_t according to (8).

$$C_t = f_t * C_{t-1} + i_t * \tilde{C} \tag{8}$$

Firstly, multiply the old state by f_t to discard some information in old state. Then, add new candidate value to old state.

4. Calculated output

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{9}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{10}$$

In (9), σ is sigmoid function, W_o and b_o is respectively weights and bias of output gate, and the result of (10) h_t is the output of current memory cell.

Bidirectional LSTM (Bi-LSTM) is a combination of two layers of LSTM networks [22]. In Fig. 3, the boxes are the LSTM cells, where representes as the output of the memory unit at the forward time t and are the output of the memory model in the backward direction at time t. A contextual semantic representation of the text can be obtained by concatenating the output of the forward sequence and the backward sequence.

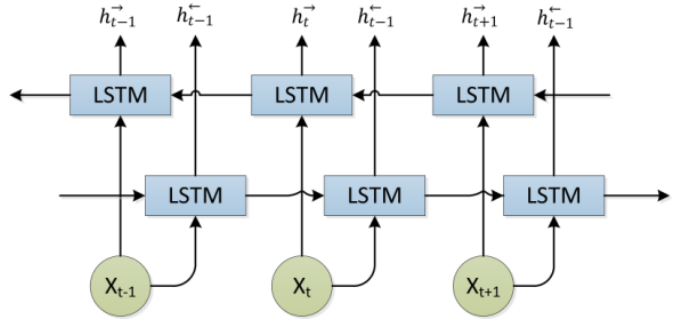


Fig. 3. Bidirectional LSTM.

C. Attention mechanism

We can take the average of the output at all times or the output at the last moment as a characteristic representation as to the output of the network [23]-[24]. However, all words contribute differently to the meaning of the sentence. Moreover, if the output of the last moment is taken as a feature representation, the previous semantic information will be lost. Therefore, to give higher weight to words that are more important to the meaning of the text, it is best to use a weighted average approach to process network output. To achieve this goal, We use the Attention mechanism to extract and aggregate words that are important to the meaning of the web page to form a text vector. Fig. 4 is a schematic diagram of the Attention mechanism. In the picture, we can see that the key to Attention mechanism is an attention matrix [25]-[26]. The calculation process of attention weights is shown in (11) (12) (13).

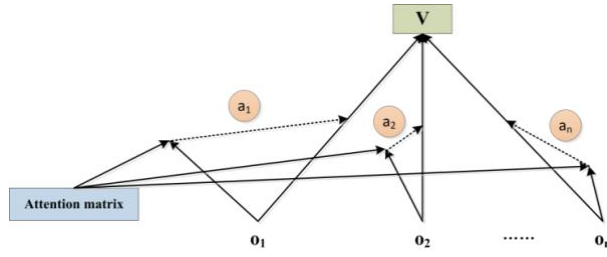


Fig. 4. Attention mechanism.

$$u_i = \tanh\left(W_o o_i + b_o\right) \tag{11}$$

$$a_i = \frac{\exp\left(u_i \cdot W_a\right)}{\sum_{i=1}^{n} \exp\left(u_i \cdot W_a\right)} \tag{12}$$

$$V = \sum_{i=1}^{n} o_i a_i \tag{13}$$

Firstly, We feed the word vector representation o_i through a one-layer MLP to get as a hidden representation of o_i according to (11). Then, as shown in (12), We use the softmax

function to obtain the standardized weight a_i of importance. Finally, according to the weight calculated in the previous step, we calculate the text vector V as the weight of each word vector. Since words that are closer to the text topic are assigned higher weights, the text vector obtains through the Attention mechanism can better express the semantics of the text.

D. Deep Web Data Source Feature Extension

Since some data sources contain a few words, feature extension is required first. Firstly, we need to extract keywords from the Deep Web data source. Generally speaking, in a sentence, verbs, and nouns usually express the meaning of sentences. Although adjectives and adverbs have no practical meaning, they appear in conjunction with nouns and verbs. Therefore, we choose these four types of words as the starting point for feature expansion [15].

Firstly, the N-gram model is used to build a feature extension library in this paper. Starting from word w_A, we can get conditional probability $P(w_B|w_A)$ of word sequence w_Aw_B, if $P(w_B|w_A) > P$, add w_B to the extension library and continue the process with w_B as the starting point until the maximum number of extensions M is reached. Detailed description is as follows:

Algorithm 1: Feature extension for Deep Web data source

Input: Original text of data source W
Threshold P of conditional probability

Extension library W_E={}
Set starting point word set W_S=W
while: Number of extensions <= M
  Starting point set of the next round of extension W'={}
  for: each word w_i in W
    for: each word sequence $w_i w_{next}^i$
      if $P(w_{next}^i|w_i) > P$
        Add w_{next}^i to the extension set W_E
        Add w_{next}^i to W'
      else
        Continue
    end for
  Number of extensions plus 1
  $W_S = W'$
  $W' = \{\}$
  end for
end while
end

Output: extension library W_E

E. Deep Web Data Source Classification Based on Attention-based Bi-LSTM

After feature extension, we solved the problem of the sparseness of feature. Then, our goal is to achieve accurate classification of data sources.

Through observation, we found that the structural difference of the data source is not very obvious and some data sources maybe only contain one or two html controls, which leads to bad performance of data source classification based on structural information. In addition, there is currently no large-scale Deep Web data source dataset that can be used to train deep networks. Based on the above questions, in this paper, we classify data by text in the Deep Web page, which not only can accurately classify the data source with simple structure but also can use large-scale text classification data set in the training phase. The core of our model present in this paper is an Attention-based Bi-LSTM model, its network structure is shown in Fig. 5. Our classification model is consist of a word embedding layer, input layer, a forward LSTM, a backward LSTM, an attention layer, a fully connected layer and a softmax classifier.

Let word sequence $W = \{w_1, w_2, \dots, w_n\}$ is all text of a Deep Web data source. Firstly, we convert W to word vector sequence $X = \{x_1, x_2, \dots, x_n\}$ by word2vec. Then, we use X as the input of forward LSTM and backward LSTM. The output sequence of the two layers of LSTM is $H^{\rightarrow} = \{h_1^{\rightarrow}, h_2^{\rightarrow}, \dots, h_n^{\rightarrow}\}$ and $H^{\leftarrow} = \{h_1^{\leftarrow}, h_2^{\leftarrow}, \dots, h_n^{\leftarrow}\}$ respectively, and the output sequence of Bi-LSTM network is $O = \{o_1, o_2, \dots, o_n\}$, in which, $o_i = [h_t^{\rightarrow}, h_t^{\leftarrow}]$. Next, we compute the weight sum of sequence O as the output of the Attention-based Bi-LSTM. Finally, obtain the classification label by fully connected layer and a softmax classifier.

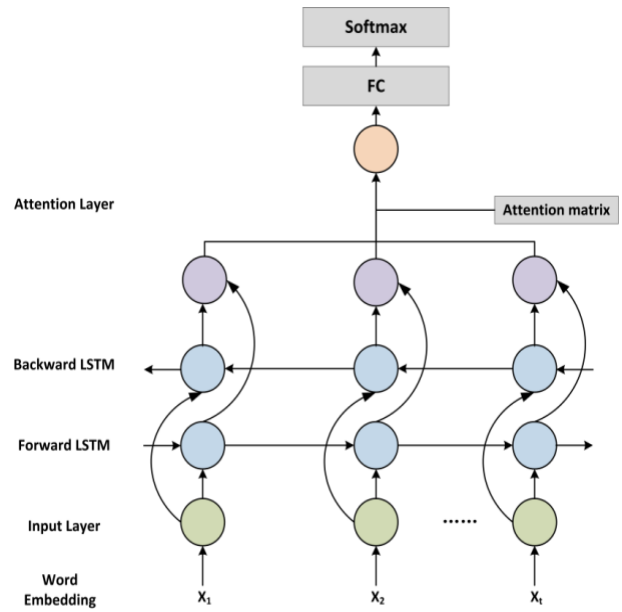


Fig. 5. Attention-based Bi-LSTM for Deep Web data source classification.

The main process of the proposed method is shown in Fig. 6 and described as follows:

(1) Since the data source of the deep web is the <form> tag in the html page, the first step is to parse the html and locate the <form> tag. In order to deal with non-standard html code and mismatched tags, we use Jsoup to parse the html code, which can fills in missing tags automatically. Moreover, in some <form> tags, some drop-down menus have hundreds of options. These drop-down menus are generally noises that do not help the classification (e.g. country, state). Therefore, drop-down menus with options greater than 30 are not used as feature.

(2) Next, we get all text between <form> tag through XML parsing and carry out some pretreatment (e.g. lemmatization, tokenize) by Stanford's CoreNLP and natural language toolkit (NLTK) [27]-[29].

(3) Then, we convert them into word vectors via Google word2vec.

(4) Build our Attention-based Bi-LSTM Deep Web data source classification.

(5) Training and test the classification model.

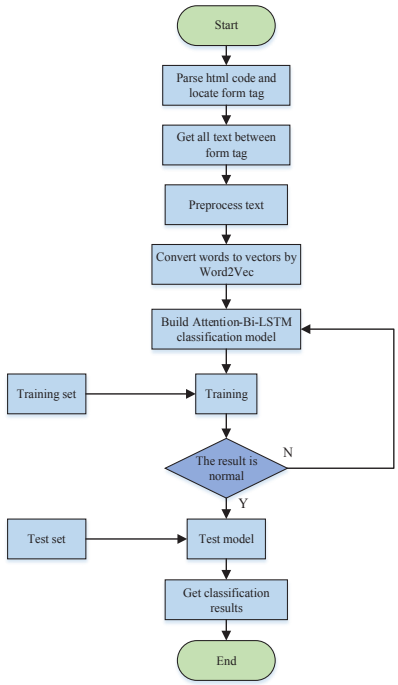


Fig. 6. The flow of our classification method.

III. EXPERIMENT AND DISCUSSION

A. DataSets

We experiment on UIUC TEL-8 datasets and evaluate the performance of our model. The UIUC TEL -8 dataset contains the original query interface of 447 Deep Web sources from 8 representative domains and its manually extract query functions. The 8 areas are further divided into three groups:(1) In the Travel group: Airfares, Hotels, and Car Rentals; (2) In the Entertainment group: Books, Movies, and Music Records; (3) In the Living group: Jobs and Automobiles.

B. Result and Analysis

1) Experiment for feature extension

In experiment for feature extension, we illustrate the effect of the feature extension process to the data source classification effect.

First of all, due to the sparseness of the data, smoothing techniques are needed in the process of training Bi-grams. We tried 4 smoothing algorithms: Add-one smoothing; Good-Turing smoothing; Interpolation; Kneser-Key smoothing. The evaluation index of the N-gram model is the perplexity, and its calculation method is as follows:

For the sentences $S = \{w_1 w_2 \cdots w_n\}$ in the test set, the perplexity is calculated as (14).

$$\begin{aligned} \mathrm{PP}(S) &= P(w_1 w_2 \cdots w_n)^{-\frac{1}{n}} \\ &= \sqrt[n]{\frac{1}{P(w_1 w_2 \cdots w_n)}} \\ &= \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_1 | w_1 w_2 \cdots w_{i-1})}} \end{aligned} \quad (14)$$

For Bi-gram,

$$\mathrm{PP}(S) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_i | w_{i-1})}} \quad (15)$$

The perplexity of model when using different smoothing algorithm is shown in Table I.

It can be seen from the above results that the selection of the smoothing algorithm is very important. According to the experimental results, we choose the Kneser-Key smoothing as the smoothing algorithm of our method.

TABLE I
THE PERPLEXITY OF MODEL WHEN USING DIFFERENT SMOOTHING ALGORITHM

Smoothing algorithm	Perplexity
Add-one smoothing	4253
Good-Turing smoothing	754
Interpolation	568
Kneser-Key smoothing	532

In the process of feature extension, the parameters that need to be manually selected are P and M. Firstly, we set M=3 and carry out text feature extension of Deep Web data source text with different P and use the Attention-based Bi-LSTM network

for feature extraction and classification. The experimental results are shown in Fig. 7.

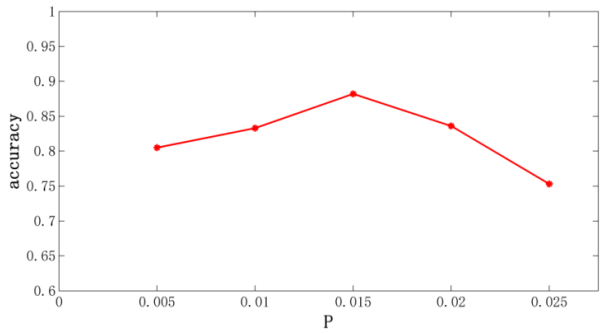


Fig. 7. The influence of the threshold P.

From the Fig.7 we can see that with the increase of P, the classification precision shows a trend of rising first and then decreasing. This is because when P is bigger, the restriction on feature extension is larger, and the fewer extension words are added to original text, so that the problem of feature sparseness cannot be solved. And when P is smaller, more extension words will be added, but the semantic difference between these words and the original text will be larger, which will bring noisy. When P=0.015, the best classification performance can be obtained.

Secondly, we set P=0.015 and change the value of M, respectively 0,1,2,3,4,5,6, where M=0 means no feature extension is performed. The results are shown in Fig. 8.

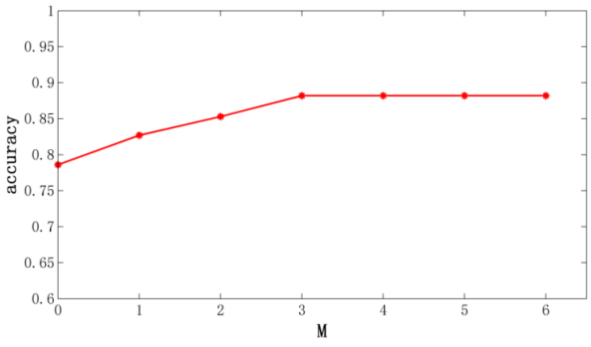


Fig. 8. The influence of maximum number of extension M.

As we can see in Fig. 8, the precision first increases with the increase of M, and then basically does not change when M>=3. Since the increase of M will increases the consumption of time, setting M=3 is the optimal choice.

Finally, we experiment to compare the performance of the classification model with and without feature expansion. The results are shown in Fig. 9.

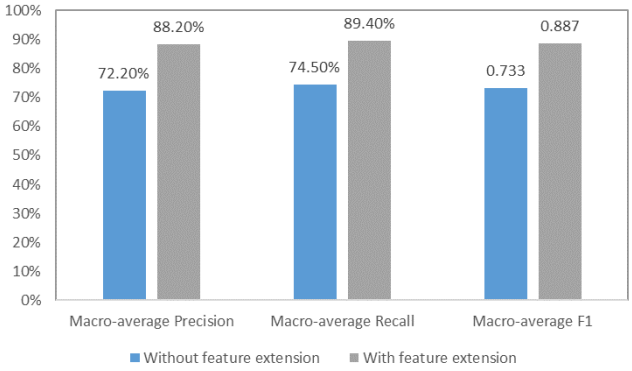


Fig. 9. Advantage of the feature extension.

Obviously, adding feature extension process can significantly improve performance of classification model.

2) Experiment for data source classification

In the second set of experiments, The performance of the Attention-based Bi-LSTM feature extraction and classification method is evaluated by comparison with existing methods. We compare our approach with the methods widely used for text classification. We choose the term frequency-inverse document frequency and support vector machine based(TF-IDF+SVM), latent dirichlet allocation and support vector machine-based(LDA+SVM), convolutional neural network (CNN) based Deep Web data source classification method as baselines. In this experiment, all results are obtained after the feature extension. The result of our experiments is shown in Fig.10.

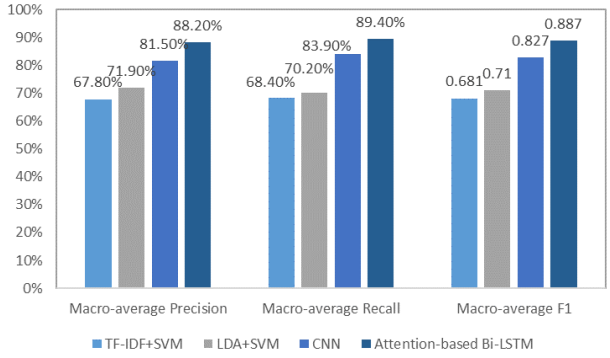


Fig. 10. The performance of different Deep Web data source classification method.

We use precision, recall and F-measure as evaluation indexes. Compare with two shallow learning methods based on SVM, our model outperforms them in precision by 20.4% and 16.3% respectively and in recall by 21.0% and 19.2%. Meanwhile, Attention-based Bi-LSTM classification model is 6.7% higher than CNN based method in precision and 5.5%

higher in recall. From Fig. 10 we can clearly see that our classification method has the best performance among the four methods.

Besides, we experiment to prove the optimization effect of the Attention mechanism on the classification model. The experimental results are shown in Fig. 11, where AVG indicates averaging and MAX indicates max-pooling.

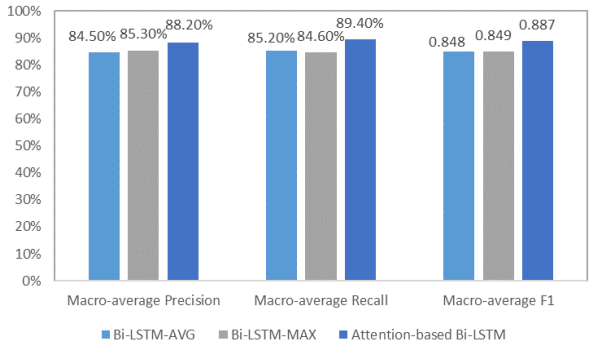


Fig. 11. Advantage of the attention mechanism.

Using the Attention mechanism to process the output at each time can obtain a more accurate vector representation of the text, thus achieving a more accurate classification of Deep Web data source. The model with Attention mechanism is 3.7% and 2.9% higher in precision and 4.2% and 4.8% higher in recall than models using averaging and max-pooling, respectively.

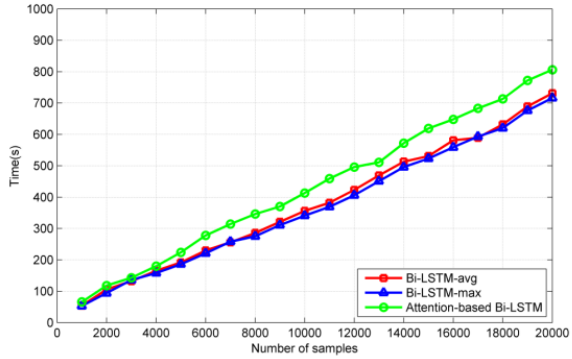


Fig. 12. Comparison of cost of time in training stage.

Finally, to prove that our classification method is feasible, we compare the network training time before and after adding the Attention mechanism when the network parameters and the number of samples are the same. In Fig. 12 we can see that the training time of model with Attention mechanism is longer than models that are not using it. However, the increase in training time is not significant. Therefore, the method we propose is completely feasible.

As can be seen from the above experimental results that the propose method is higher in precision than the existing method, and at the same time, there is no great increase in time consumption during the training stage.

IV. CONCLUSIONS

In this paper, we propose a Deep Web data source classification method based on text feature extension and extraction. In text feature extension stage, to solve the problem of feature sparseness, we built the feature extension using the N-gram model. In the Deep Web data source classification stage, we build a classification model based on Attention-based Bi-LSTM. Moreover, the Attention mechanism can give greater weight to words that are more relevant to the category of text, so that more accurate text vector representation can be obtained. The experimental results not only show that our model has significant advantages over the previous method, especially for Deep Web data sources with less text content, but also prove that the use of the Attention mechanism can improve the precision without a huge increase in the cost of training time. In conclusion, the Deep Web data source classification method based on text feature extension and extraction is a high performance and feasible method.

REFERENCES

[1] Lopez-Sanchez, Daniel, Gonzalez Arrieta, Angelica, Corchado, Juan M, "Visual content-based web page categorization with deep transfer learning and metric learning," NEUROCOMPUTING, pp. 418-431, 2019. DOI: 10.1016/j.neucom.2018.08.086.

[2] Feng Zhao, Jingyu Zhou, Chang Nie, Heqing Huang, Hai Jin, "SmartCrawler: A Two-Stage Crawler for Efficiently Harvesting Deep-Web Interfaces," IEEE Transactions on Services Computing, vol. 9, no. 4, pp. 608-620, 2016. DOI: 10.1109/TSC.2015.2414931.

[3] Marin Castro, Heidy Marisol, Sosa Sosa, Victor, Nuno Maganda, "Automatic construction of vertical search tools for the Deep Web," IEEE Latin America Transactions, 2018, vol. 16, no. 2, pp. 574-584, Feb. 2018. DOI: 10.1109/TLA.2018.8327415.

[4] Hernandez, Inma, Rivero, Carlos R, Ruiz, David, "Deep Web crawling: a survey," WORLD WIDE WEB-INTERNET AND WEB INFORMATION SYSTEMS, pp. 1577-1610, 2019. DOI: 10.1007/s11280-018-0602-1.

[5] Yuan, Jing, et al, "Result Merging for Structured Queries on the Deep Web with Active Relevance Weight Estimation," Information Systems, 64:93-103, 2017. DOI: 10.1016/j.is.2016.06.005.

[6] Barrio, Pablo, L. Gravano, "Sampling strategies for information extraction over the deep web," Information Processing & Management, 53.2:309-331, 2017. DOI: 10.1016/j.ipm.2016.11.006.

[7] Prafful Mishra, "Accuracy Crawler: An Accurate Crawler for Deep Web Data Extraction," 2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCCT), 2018. DOI: 10.1109/ICCPCCT.2018.8574286.

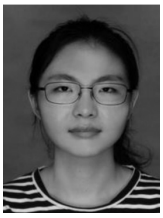
[8] Ye Hongfan, Cao Buqing, Peng Zhenlian, "Web Services Classification Based on Wide & Bi-LSTM Model," IEEE ACCESS, 43697-43706, 2019. DOI: 10.1109/ACCESS.2019.2907546.

[9] Z. Wang, Z. Qu, "Research on Web text classification algorithm based on improved CNN and SVM," 2017 IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, pp. 1958-1961, 2017. DOI: 10.1109/ICCT.2017.8359971.

[10] Fatih Ertam, "Deep learning based text classification with Web Scraping methods," 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), 2018. DOI: 10.1109/IDAP.2018.8620790.

[11] Serkan Ballı, Onur Karasoy, "Development of content-based SMS classification application by using Word2Vec-based feature extraction," IET Software, Volume: 13, Issue: 4, 2019. DOI: 10.1049/iet-sen.2018.5046.
[12] Shi M, Tang Y, Liu J, "Functional and Contextual Attentionbased LSTM for Service Recommendation in Mashup Creation," IEEE Transactions on Parallel and Distributed Systems, pp:1-1, 2018. DOI: 10.1109/TPDS.2018.2877363.
[13] Yanliang Jin, Can Luo, Weisi Guo, Jinfei Xie, Dijia Wu, Rui Wang, "Text Classification Based on Conditional Reflection," IEEE Access, Volume: 7, pp: 76712-76719, 2019. DOI: 10.1109/access.2019.2921976.
[14] Yangsen Zhang, Jia Zheng, Yuru Jiang, Gaijuan Huang, Ruoyu Chen, "A Text Sentiment Classification Modeling Method Based on Coordinated CNN-LSTM-Attention Model," Chinese Journal of Electronics, volume 28, pp: 120-126, 2019, DOI: 10.1049/cje.2018.11.004.
[15] B. Sun, P. Zhao, "Feature extension for Chinese short text classification based on topical N-Grams," 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), Wuhan, pp. 477-482, 2017. DOI: 10.1109/ICIS.2017.7960039.
[16] Xinwei Zhang, Bin Wu, "Short Text Classification based on feature extension using The N-Gram model," 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, pp. 710-716, 2015. DOI: 10.1109/FSKD.2015.7382029.
[17] Yang, Zichao, et al, "Hierarchical Attention Networks for Document Classification," Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1480-1489, 2017.DOI: 10.18653/v1/n16-1174.
[18] Ullah, J, Ahmad, K, Muhammad, M. Sajjad, S. W. Baik, "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features," IEEE Access, vol. 6, pp. 1155-1166, 2018. DOI: 10.1109/ACCESS.2017.2778011.
[19] L. Gao, Z. Guo, H. Zhang, X. Xu, H. T. Shen, "Video Captioning With Attention-Based LSTM and Semantic Consistency," IEEE Transactions on Multimedia, vol. 19, no. 9, pp. 2045-2055, 2017. DOI: 10.1109/TMM.2017.2729019.
[20] K. Fu, J. Jin, R. Cui, F. Sha, C. Zhang, "Aligning Where to See and What to Tell: Image Captioning with Region-Based Attention and Scene-Specific Contexts," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2321-2334, 2017. DOI: 10.1109/TPAMI.2016.2642953.
[21] Y. Wang, P. Yu, H. Li, H. Li, "Research on the Recognition of Offline Handwritten New Tai Lue Characters Based on Bidirectional LSTM," International Conference on Network, Communication, Computer Engineering, 2018. DOI: 10.2991/ncce-18.2018.189.
[22] Kim B, Chung K, Lee J, Seo J, Koo M.-W, "A Bi-LSTM memory network for end-to-end goal-oriented dialog learning," Computer Speech & Language, 53, 217-230, 2019. DOI: 10.1016/j.csl.2018.06.005.
[23] W. Du, Y. Wang, Y. Qiao, "Recurrent Spatial-Temporal Attention Network for Action Recognition in Videos," IEEE Transactions on Image Processing, vol. 27, no. 3, pp. 1347-1360, 2018. DOI: 10.1109/TIP.2017.2778563.
[24] F. Fahimi, C. Guan, W. B. Goh, K. K. Ang, C. G. Lim, T. S. Lee F. Fahimi, "Personalized features for attention detection in children with Attention Deficit Hyperactivity Disorder," 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Seogwipo, pp. 414-417, 2017. DOI: 10.1109/EMBC.2017.8036850.
[25] Zhang Y, Gao X, Peng X, Ye J, Li X, "Attention-Based Recurrent Temporal Restricted Boltzmann Machine for Radar High-Resolution Range Profile Sequence Recognition," Sensors, 18, 1585, 2018. DOI: 10.3390/s18051585.
[26] Shen C, Huang T, Liang X, Li F, Fu K, "Chinese Knowledge Base Question Answering by Attention-Based Multi-Granularity Model," Information, 9, 98, 2018. DOI: 10.3390/info9040098.
[27] M. Ali, S. Khalid, M. H. Aslam, "Pattern-Based Comprehensive Urdu Stemmer and Short Text Classification," IEEE Access, vol. 6, pp. 7374-7389, 2018. DOI: 10.1109/ACCESS.2017.2787798.
[28] N. Kumar Nagwani, "A Comment on "A Similarity Measure for Text Classification and Clustering," IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 9, pp. 2589-2590, 2015. DOI: 10.1109/tkde.2015.2451616.
[29] C. Liu, W. Hsaio, C. Lee, T. Chang, T. Kuo, C. Liu, "Semi-Supervised Text Classification With Universum Learning," IEEE Transactions on Cybernetics, vol. 46, no.

Yuancheng Li, received the Ph.D degree from University of Science and Technology of China, Hefei, China, in 2003. From 2004 to 2005, he was a postdoctoral research fellow in the Digital Media Lab, Beihang University, Beijing, China. Since 2005, he has been with the North China Electric Power University, where he is a professor and the Dean of the Institute of Smart Grid and Information Security. From 2009 to 2010, he was a postdoctoral research fellow in the Cyber Security Lab, college of information science and technology of Pennsylvania State University, Pennsylvania, USA.



Guixian Wu was born in 1994 in guizhou Province, China. Since 2018, she has been a master's student in computer science and technology at north China electric power university in Beijing, China. Her research interests include text processing and artificial intelligence applications.

Xiaohan Wang was born in 1993 in Beijing, China. He received B.S. degree in information security from North China Electric Power University, Beijing, in 2016. Since 2016, he is a M.S. candidate in computer science and technology at North China Electric Power University. His research interests include natural language