Semantic Information Detection of Webpage Based on Word Vector and Infomap

Yuqian Wang
Information Center
China Institute of Atomic Energy
Beijing, China
yqwang7@163.com

Jianyou Lv
Information Center
China Institute of Atomic Energy
Beijing, China
ljy@ciae.ac.cn

***Abstract*—For Chinese web pages, we use regular expression and Viterbi algorithm to realize Chinese filtering and word segmentation, then use ngram2vec algorithm to get the word vector set of web page and pre train the word vector set of Baidu Encyclopedia. Baidu Encyclopedia word vector set is based on Infomap clustering algorithm to realize word vector Clustering and tagging types, training neural network through training data set and Baidu Encyclopedia corpus to determine the type of unknown web pages through neural network, and achieve the purpose of detecting the semantic information of unknown web pages. This algorithm is has few super parameters and high calculation efficiency. Experiments show that the accuracy of the trained neural network model reaches 96.73%, which can quickly and accurately identify the type of web page.**

Keywords—Web Semantic Detection, Web classification, word vector, Infomap; multilayer neural network BP algorithm

I. INTRODUCTION

With the development of the information age, in our daily life and work, we use script language and browser plug-ins to simplify our life, which brings convenience and speed, but there may also be the risk of information disclosure, often by phishing attacks. On the other hand, in order to improve the overall retrieval ability and improve the satisfaction of users, it is necessary to detect the web information and classify the content that users have browsed, and make relevant recommendations. Therefore, no matter from the perspective of network security or the direction of commercial value, web semantic information detection has a great use, so in recent years, related algorithms are emerging in an endless stream, and the requirements for the accuracy of detection are naturally higher and higher. At present, the common web page detection methods are KNN algorithm and naive Bayes algorithm, but the detection of Chinese Web page has not been a good solution. Therefore, for Chinese web pages, we use skip gram model to transform Chinese into word vector, and combine Infomap clustering algorithm and neural network BP algorithm to train a neural network model that can determine the type of unknown Chinese web pages. Finally, we analyze and predict the unknown web pages included in the experiment set.

II. METHOD

First of all, crawling a variety of initial seed files to get the initial data and filtering out the Chinese text, then using regular expression to get the text data and realizing the Chinese word segmentation by hidden Markov model and Viterbi algorithm. Then we transform the Chinese word into word vector using neural network language model, and carry out the pre training on the corpus of Baidu Encyclopedia. The word vector set with a word bag capacity of more than one million dimensions and 256 dimensions is obtained. On this basis, the word vector set of Baidu Encyclopedia implements word vector clustering and type labeling based on Infomap clustering algorithm. On this basis, neural network is trained by combining training data set, and the unknown web page type is determined by neural network, as shown in the figure below:

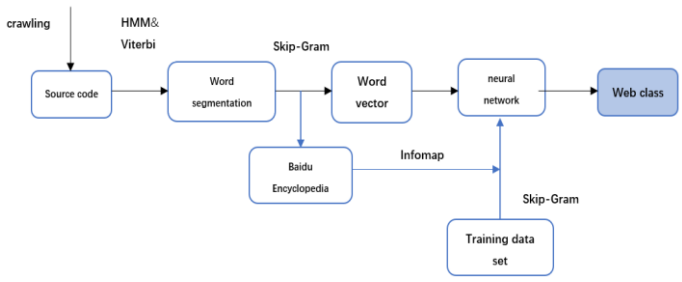

Fig. 1. Task Flow Chart.

A. HMM an Viterbi algorithm

HMM model is a statistical analysis model based on probability. It has three basic assumptions:

Hypothesis 1: Markov hypothesis (limited historical hypothesis)

$$p\left(x_i \mid x_{i-1} \ldots x_1\right) = p\left(x_i \mid x_{i-1}\right) \tag{1}$$

Hypothesis 2: Homogeneity hypothesis (state independent of specific time)

$$p\left(x_{i+1} \mid x_i\right) = p\left(x_{j+1} \mid x_j\right) \tag{2}$$

Hypothesis 3: Output independence hypothesis (output is only related to the current state)

$$p\left(o_1 \ldots o_T \mid x_1 \ldots x_T\right) = \prod p\left(O_t \mid x_t\right) \tag{3}$$

The application of HMM in Chinese word segmentation is realized by using word segmentation as the sequence annotation task of words in string: each word will occupy a certain word formation position when forming a specific word: B (the beginning of the word), M (the middle of the

word), E (the end of the word), S (the word is formed separately). Here, assuming $\lambda_1\lambda_2....\lambda_n$ is the visible state chain in the HMM, and $o_1o_2...o_n$ is the implicit state chain in the model, where n represents the number of words in the sentence, λ represents the word, o represents one of the four positions. Requiring the max probability of $(o_1o_2...o_n \mid \lambda_1\lambda_2...\lambda_n)$.

Based on three basic assumptions of HMM and Bayesian formula, we can get (4):

$$p(\lambda \mid O)p(O) \sim p(\lambda_1 \mid o_1)p(o_2 \mid o_1)p(\lambda_2 \mid o_2)p(o_3 \mid o_2)...p(o_n \mid o_{n-1})p(\lambda_n \mid o_n) \tag{4}$$

Where $P(\lambda_i \mid o_i)$ is the output probability, $P(o_i \mid o_{i-1})$ is the conversion probability, and set $P(\mathrm{B} \mid \mathrm{B}) = 0$, $P(\mathrm{M} \mid \mathrm{E}) = 0$, $P(\mathrm{B} \mid \mathrm{M}) = 0$, $P(\mathrm{X} \mid \mathrm{S}) = 0$(X=M,E),which are to exclude the occurrence of BB, EM and other unreasonable situations.

According to statistical corpus, the frequency in the initial probability calculation is the number of states corresponding to the first word of each sentence; the frequency in the output probability calculation is the number of words corresponding to the corresponding states. The frequency of shift probability calculation is to count the number of states from each word (i.e. BMES) to the next word. Viterbi algorithm is used to find out the hidden state chain, which is the final annotation result. Its central idea is that if the final optimal path passes o_i through a certain path, then the path from the initial node to the point o_{i-1} is also an optimal path. The Viterbi algorithm can only proceed to the next step after calculating the optimal choice in each step. After all the steps are completed, the optimal path is obtained by backtracking.

The overall calculation steps are as follows:

1) Choosing Chinese corpus of Wikipedia as training set data;

2) Obtaining the initial probability, output probability and transfer probability of HMM by statistical corpus information;

3) Getting the optimal path by Viterbi algorithm, which is the segmentation result.

B. Skip-Gram Model Based on Hierarchical Softmax

The data set of the set of words is got through the HMM and Viterbi algorithm. In natural language processing, words need to be transformed to the form of vector for computer identification. The earliest way of word vector coding is one-hot coding, but it will generate the problem of dimension disaster with the continuous growth of the capacity of the dictionary. Distributed representation can not only associate words with each other, but also smoothly transition the vectors to the function model of probability to get continuous dense vectors. Before the appearance of word2vec, DNN was used to train word vector and deal with the relationship between words. However, the efficiency of DNN in dealing with large capacity vocabulary is too low, so the optimized neural language model −− skip-gram model based on hierarchical softmax is adopted. Our ultimate goal is to get the optimal model parameters, namely the weight matrix.

Skip-gram model is used to predict the context words corresponding to a given central word, as shown in the figure below:

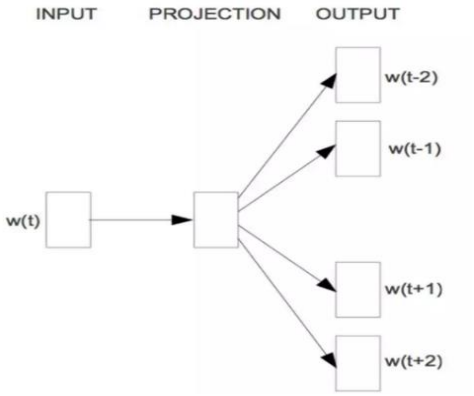

Fig.2. Skip-Gram Model.

In the figure above, the central word $w(t)$ is the given input word, which will enter the hidden layer and the weight matrix for point product calculation. The calculated dot product results are transferred to the output layer, and then the dot product is calculated again with the weight matrix of the output layer. Finally, the probability of appearing in the specified context is calculated by using the softmax activation function.

The probability function is as follows:

$$p(w_o \mid w_i) = \frac{e^{U_o \cdot V_i}}{\sum_j e^{U_j \cdot V_i}} \tag{5}$$

Where U and V respectively represent the output vector and input vector of the word of the central word.

It can be seen from the above that from the hidden layer to the softmax layer, the calculated amount of the softmax probability of all words and sort them is huge, so word2vec uses the Huffman tree instead of the mapping from the hidden layer to the output softmax layer to avoid calculating the probability of all words. All the internal nodes of the Huffman tree are similar to the neurons in the hidden layer of the previous neural network, in which the word vector of the root node corresponds to projected word vector, and all the leaf nodes are similar to the neurons in the output layer of the previous neural network softmax, and the number of leaf nodes is the size of the vocabulary. Since the Huffman tree is a binary tree, the amount of calculation has changed from M to $\log_2 M$ (assuming the vocabulary is M). In addition, the characteristics of the Hoffman tree show that the more high-frequency words are closer to the root of the tree, the easier they are to be found, which also conforms to the principle of the minimum entropy in information theory.

Adopting the method of binary logic regression, which

stipulates that walking along the right subtree is positive, Huffman code is 0, otherwise it is negative, Huffman code is 1, and the weight of the left subtree is not less than that of the right subtree. The decision method is sigmoid function:

$$P(+)=\sigma\left(x_w^T\theta\right)=\frac{1}{1+e^{-x_w^T\theta}} \tag{6}$$

$$P(-)=1-\sigma\left(x_w^T\theta\right)=\frac{e^{-x_w^T\theta}}{1+e^{-x_w^T\theta}} \tag{7}$$

Where x_w is the word vector of the hidden layer, and θ is the model parameter that we need to get from the training sample.

The hierarchical softmax function uses the maximum likelihood method to find the word vectors of all nodes and the parameters of all internal nodes. For a central word, maximizing the following likelihood functions:

$$\prod_{i=2}^{l_w} P\left(d_i^w \mid x_w,\theta_{i-1}^w\right)=\prod_{i=2}^{l_w}\left[\sigma\left(x_w^T\theta_{i-1}^w\right)\right]^{1-d_i^w}\left[1-\sigma\left(x_w^T\theta_{i-1}^w\right)\right]^{d_i^w} \tag{8}$$

Where, l_w represents the total number of nodes in the path from the root node to the leaf node, p_i is the i_{th} node of the path w, the corresponding Huffman code is $d_i^w \in \{0,1\}$ and $i=2,3,...l_w$, and the corresponding model parameter is θ_i^w and $i=2,3,...l_w-1$.

Using the random gradient rising method to solve the gradient expression of model parameters first, and then iterate step by step to solve x_w and θ_i^w that we need, that is, the word vector and model parameters.

C. "Community Discovery" Algorithm—Infomap Algorithm

The so-called "community discovery" is used to discover the community structure in the network, which can also be regarded as a clustering algorithm. Infomap algorithm is a kind of unsupervised algorithm, which does not need to set the number of clustering centers and other parameters, and has high efficiency and expansibility. From information theory we see that Huffman coding can make the average coding length the shortest, and the shortest average coding length is information entropy. Infomap algorithm is used to minimize the objective information and the learning cost, so it embodies the principle of minimum entropy.

In Infomap algorithm, clustering objects are regarded as all nodes in a directed graph. There is an edge between each node in the graph, and each edge has a weight, that is, a transition probability. Jumping from one point to another by "random walk", and then jump from this node to the next. Repeating this process until reaching all the nodes to get the transition probability and construct Huffman code. After "random jump", a very long sequence of nodes has been gotten. In order to effectively compress information, hierarchical coding is used to calculate the average coding length, so that the hierarchical coding scheme with the shortest average coding length is the clustering result. The algorithm flow is as follows:

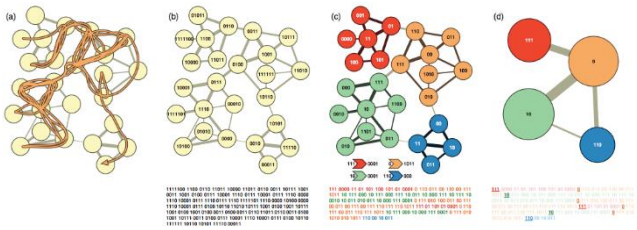

Fig.3. Infomap Algorithm Flow.

In practical application, it is not necessary to generate sequence by "random walk" but to know the probability and transfer probability of each object in the "random walk" generation sequence. Regarding word vectors as nodes in digraph, and the transfer probability between nodes and nodes can be expressed by cosine distance normalization between vectors. According to the flow of Infomap algorithm, the minimum average coding length is composed of the average coding length within a class and the average coding length between classes:

$$L(M)=q_{\curvearrowright}\left(-\sum_i \frac{q_{i\curvearrowright}}{q_{\curvearrowright}}\log\frac{q_{i\curvearrowright}}{q_{\curvearrowright}}\right)+\sum_i p_{i\circlearrowright}\left(-\frac{q_{i\curvearrowright}}{p_{i\circlearrowright}}\log\frac{q_{i\curvearrowright}}{p_{i\circlearrowright}}-\sum_{\alpha\in i}\frac{p_\alpha}{p_{i\circlearrowright}}\log\frac{p_\alpha}{p_{i\circlearrowright}}\right) \tag{9}$$

$$p_{i\circlearrowright}=q_{i\curvearrowright}+\sum_{\alpha\in i}p_\alpha \tag{10}$$

$$p_\alpha=\sum_\beta p_\beta p_{\beta\to\alpha} \tag{11}$$

$$q_{i\curvearrowright}=\sum_{\beta\in i}\sum_{\alpha\notin i}p_\beta p_{\beta\to\alpha} \tag{12}$$

Where p_α is the probability of node α occurrence, $q_{i\curvearrowright}$ is the probability of category i occurrence, $p_{i\circlearrowright}$ is the sum of node probability and category probability within the category.

The whole Infomap algorithm flow is as follows:

1) The cosine distance between word vectors is calculated and normalized to be the transition probability;

2) The probability of node and category is calculated by transfer probability;

3) The shortest average coding length is obtained by traversing the search clustering scheme.

III. Experiment

Training Baidu Encyclopedia word vector set with a capacity of more than one million through word2vec tool, and cluster word vectors through Infomap algorithm to get

different "word vector communities".

First ,choosing a variety of types of Chinese Web page data as the starting seed files, then using the crawler to crawl the relevant web sites in the starting seed files to get a series of Web site data sets with category identification, which are mainly divided into 9 categories: music, blog, e-commerce, education, science, government affairs, post bar, video and game, the crawling of music website is as follows:

```
# ---------------------------- MUSI ----------------------------
MUSI https://y.qq.com/
MUSI https://music.163.com/
MUSI http://www.kugou.com/
MUSI http://bd.kuwo.cn/
MUSI https://www.xiami.com/
MUSI http://music.migu.cn/
```

Fig.4. Start Seed File.

```
MUSI https://y.qq.com/n/yqq/song/000l25YK3JR0Lu.html
MUSI https://y.qq.com/n/yqq/playlist/2657525475.html#stat=y_new.song.hotgedan.click
MUSI https://y.qq.com/portal/playlist.html#t3=1&t2=5&t1=167
MUSI https://y.qq.com/n/yqq/playlist/7085583499.html#stat=y_new.playlist.dissname
MUSI https://y.qq.com/n/yqq/song/001s3IZ91a0H5Y.html
MUSI https://y.qq.com/artists
MUSI https://y.qq.com/n/yqq/song/001s3IZ91a0H5Y.html#comment_box
```

Fig.5. Website Data Crawled By Crawler.

After getting the web address data set with category identification, crawling the web address to get the source code and filtering out Chinese by regular expression, then saving the source code and Chinese locally. In addition, TF-IDF technology is applied to select five words with the highest importance in each web site. The central idea of TF-IDF is to find out the words that appear more frequently in this text than in all texts. The expression is as follows:

$$TF-IDF_w = TF_w \times IDF_w = \frac{N_w}{N}\log\left(\frac{Y}{Y_m+1}\right) \quad (13)$$

Where N_w and Y_m respectively represent the number of times that entries appear in the text and the number of documents that contain w in the corpus, N and Y respectively represent the total number of words in the text and the total number of documents in the corpus.

The first five representative words are got by comparing $TF-IDF_w$. Their word vectors will be directly found in the Baidu Encyclopedia word vector set. If the set does not contain the corresponding word vectors, finding the corresponding "word vector community" and getting the similar word vectors. In this way, the matrix representing the Chinese website will be completed. Then these matrix datasets with category identification will be taken as training data set to train the neural network. In order to reduce the complexity, combination of the sentence vector by averaging 5 word vectors and the first word vector acts as the input of the neural network, and the type of web address acts as the output. In addition, relu function and cross entropy function are set as activation function and loss function respectively, and Adam optimizer is used to accelerate convergence and improve accuracy. The prediction model with 23497 super parameters is obtained by 30 iterations with an average training period of 0.73s through back propagation algorithm. Its accuracy is as follows:

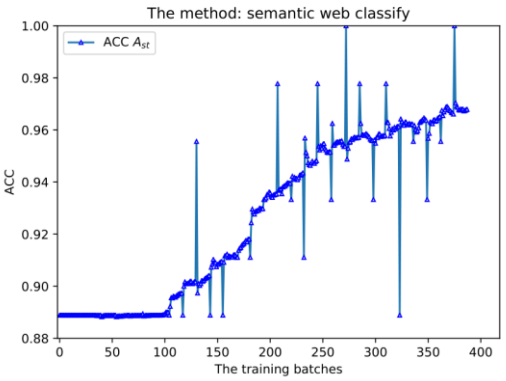

Fig.6. Accuracy Vector Diagram of Training Neural Network.

2000 pieces of data are divided into two parts: training set and experiment set, accounting for 60% and 40% respectively. KNN algorithm, naive Bayes algorithm and neural network model based on word vector are used for classification, and the accuracy of the results is as follows:

TABLE I. COMPARISON OF ACCURACY OF VARIOUS ALGORITHMS

Web page Classification Algorithm	Accuracy Rate
KNN Algorithm	92.89%
Naive Bayes Algorithm	93.76%
Neural Network Model Based on Word Vector	96.73%

It can be seen from the table that the accuracy of the training neural network based on the word vector and Infomap algorithm is 96.73%. Compared with KNN algorithm and naive Bayes algorithm, the effect of improving the accuracy is still obvious. However, there are many website classes which are different from the web address categories we define. We can continue to join different types of websites for training and continuous improvement in future.

IV. DISCCUSION

This paper proposes a Chinese Web page Semantic Detection Algorithm Based on word vector, Infomap algorithm and multi-layer neural network. Starting from the source code of Web site, the Chinese information of web page is obtained and further transformed into word vector that can be recognized by computer. By combining with "word vector community" in Baidu Encyclopedia corpus, a matrix representing the semantic information of Web site is obtained to train BP neural network, Thus, we can identify the unknown URL categories by ourselves, and get the accuracy of > 95%.

In order to reduce the computational complexity, a three-layer neural network with two inputs and one output is used in the whole experimental process, and the processing of neural network training data set is also relatively simplified. The accuracy can be improved by increasing the dimension of hidden layer and ISF embedding method. The future work can also take the user's browsing frequency, staying time and other directions as the reference basis to develop an accurate web page recommendation scheme.

REFERENCES

[1] Kandola, E. J. , Hofmann, T. , Poggio, T. , & Shawe-Taylor, J. . (2006). A neural probabilistic language model. Studies in Fuzziness & Soft Computing, 194, 137-186.

[2] Le, Q. V. , & Mikolov, T. . (2014). Distributed representations of sentences and documents.

[3] Bollegala, D. , Mohammed, A. , Maehara, T. , & Kawarabayashi, K. I. . (2015). Joint word representation learning using a corpus and a semantic lexicon.

[4] Newman, & M., E. J. . (2013). Spectral methods for community detection and graph partitioning. Physical Review E, 88(4), 042822.

[5] Girvan, M., Newman, & E., J. . (2002). Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America.

[6] Raghavan, U. N. , Albert, Réka, & Kumara, S. . (2007). Near linear time algorithm to detect community structures in large-scale networks. Physical Review E, 76(3), 036106.

[7] Gulikers, L. , Lelarge, M. , & Laurent Massoulié. (2015). A spectral method for community detection in moderately-sparse degree-corrected stochastic block models. Advances in Applied Probability, 49(3).

[8] Mikolov, T. , Sutskever, I. , Chen, K. , Corrado, G. , & Dean, J. . (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26, 3111-3119.

[9] Dunlop, M. , Groat, S. , & Shelly, D. . (2010). GoldPhish: Using Images for Content-Based Phishing Analysis. Internet Monitoring and Protection (ICIMP), 2010 Fifth International Conference on. IEEE.

[10] Sharma Kartik; Aggarwal Ashutosh; Singhania Tanay; Gupta Deepak; Khanna Ashish (2019). Hiding Data in Images Using Cryptography and Deep Neural Network. Journal of Artificial Intelligence and Systems, 1, 143–162.