Research on computer application software monitoring data processing technology based on NLP

J Wang¹, B J Liu¹, W He¹, J K Xue¹ and X Y Han¹
¹ China Institute of Marine Technology and Economy, Beijing, China

E-mail:w_jing1994@163.com

Abstract. In the process of using computer application software, the system may fail because of software defects. During the operation of the computer system, the running state of the system is also monitored. This paper mainly studies the monitoring data processing technology of computer application software based on NLP and presents a software monitoring data processing scheme based on NLP. First, the data were preprocessed by word segmentation and data cleaning. Then the named entity recognition and extraction information are accomplished by fusing CRF and rule-based methods. It can provide effective data support for system failure prediction by characterizing the software monitoring data.

1. Introduction
Monitoring data of computer application software refers to the data representing the running state of the system produced in the process of computer software operation and the exceptions to these data are the main characteristics of software bug triggers. The storage of software monitoring data is mainly the log files generated by the system. It is of great significance to effectively extract the monitoring data of system application software from the system log file and analyze the influence of the interaction between different parameters on software defects.

Natural language processing (NLP)[1] is a technology that transforms the language used in human communication into a machine language that can be understood by machines. It is also a technology that transforms unstructured data into structured data. There are few applications for monitoring data processing using NLP technology. This paper presents a system application software monitoring data processing scheme based on NLP technology. NLP technology is used to extract the key data from the software monitoring data, and the data is structured to provide data input for the prediction of software defects.

2. Theories related to NLP
2.1. Main methods and models of NLP
NLP as a Language used for human communication into machines can understand the Language of technology. Its main application fields include: information retrieval, machine translation, document classification, automatic question answering system, information filtering, abstract, information extraction[2], text mining, public opinion analysis, machine writing, speech recognition, etc. The purpose of information extraction is to directly obtain and record the facts and information that users are interested in from the natural language text, and to structurally process all kinds of knowledge and information contained in the unstructured text, so as to turn it into a structured organization for storage and processing.
The main methods of NLP are rule-based method and statistics-based method respectively. The rule-based method mainly relies on the method of manual rule compilation by domain experts, so that the system can deal with the information extraction problem of specific domain. However, this rationalist approach requires the intervention of domain experts, and such experts are hard to find in reality, so the development process can sometimes waste a lot of time and a lot of manpower. The method based on statistics mainly analyses the data through large-scale database, so as to realize the processing of natural language. The method based on statistics is faster than the method based on rules, but the number of training corpus needs to be large enough and the labeling is completely correct to ensure that the information extraction results can reach the required accuracy and recall rate. Therefore, in the practical use of the process is often combined with the two methods, complementary advantages and disadvantages.

The system application software monitoring data processing based on NLP mainly carries out information extraction on the collected software state data, characterizes the extracted data, and obtains the relationship between software state information and fault state through model training, so as to realize the prediction of software defects.

2.2. The process of NLP

The process of NLP is shown in figure 1.

![Figure 1. The process of NLP](image)

The process of NLP can be roughly divided into the following five steps:

Step1: The acquisition of corpus. It's basically getting the raw data.

Step2: The preprocess of the corpus. It mainly includes the steps of corpus clearing, word segmentation, part of speech tagging and stopping words.

Step3: Characterizing, also known as vectorization. It mainly represents the words and words after word segmentation into computer-computable types (vectors), which is helpful to better express the similarity between different words.

Step4: Model training. It mainly includes traditional supervised, semi-supervised and unsupervised learning models, which can be selected according to different application requirements.

Step5: Model evaluation. The model was evaluated by NLP evaluation index.

The commonly used measurement indexes of NLP include Precision, Recall and F-measure. Precision is to measure the accuracy of the retrieval system. Recall rate is to measure the recall rate of retrieval system. F-measure is the comprehensive accuracy and recall rate used to reflect the overall index. When F value is high, the test method is effective.

3. Application software monitoring data processing technology based on NLP

3.1. Overall structure

The overall architecture of system application software monitoring data processing technology based on NLP is shown in figure 2. Application software monitoring data is divided into Chinese text and English text, mainly including process running status information, occupied system resources information, interface data of cross-linking equipment. The original data is cleaned after word segmentation, and then the data is standardized. The standardized data is classified by model training. The data at the end of classification is combined with the defect state, and the software defect prediction model is obtained by machine learning algorithm. Finally, Precision, Recall rate and F-measure of commonly used evaluation indexes of NLP were used to evaluate the generated model.
3.2. Data preprocessing

Data preprocessing is mainly about word segmentation and data cleaning of text. The treatment methods adopted in this paper are as follows.

3.2.1. Text segmentation

English text segmentation can be divided by space, in the software tool through the `split()` function. There is no space between words in Chinese text, so Chinese text segmentation is an important step in text preprocessing. Among the current popular word segmentation tools, Jieba Chinese word segmentation tool is the most widely used. The process of word segmentation using Jieba tool in this paper is shown in figure 3.

The main algorithm of Jieba Chinese word segmentation is as follows:

1) In order to efficiently scan out any combination of words in the text sentences to be sliced, a directed acyclic graph DAG for this string is constructed based on the prefix dictionary.

2) The path planning method is used to query the shortest path to find the maximum segmentation combination based on word frequency statistics.

3) Based on the ability of Chinese characters to form words, Hidden Markov Mode (HMM)[3] model is used for the unlogged words, and use Viterbi algorithm to calculate.

4) TF-idf and TextRank machine learning algorithm are adopted to extract keywords of text content to be shred;

5) Viterbi algorithm is used for part of speech tagging.
3.2.2. Data Cleaning
Data cleaning is mainly to remove the non-text part of the data. In Chinese text mainly refers to the removal of interjection, question words, modal words and so on. In English text, it mainly refers to removing useless labels, special symbols, stop words and so on.

English data cleaning can be done with Natural Language Toolkit (NLTK). NLTK is a Python toolkit commonly used in NLP domain. It can remove English stop words according to the English stop word library in NLTK. But the kit does not support Chinese. Therefore, Chinese stop words can be constructed according to the commonly used Chinese stop words list (including 1208 stop-words) and then removed by NLTK.

3.2.3. Stemming and Lemmatization
The English text at the end of data cleaning also needs to be extracted from the word stem and restored to the word form. Stemming is the extraction of the stem or root of a word. Lemmatization is to reduce any form of language vocabulary into a general form. NLTK provides many methods for word stem extraction and form reduction. In this article, wordnet is used without oversimplifying the words.

3.3. A fusion method of CRF and Rule - based named entity recognition and extraction.
In this paper, a named entity recognition and extraction method combining Conditional Random Fields (CRF)[4] and Rules-based method is used to extract the features of the text of the software monitoring data. Firstly, the named entity is recognized based on the CRF model, and then the second recognition and extraction is carried out by the rule-based method.
3.3.1. Named entity recognition based on CRF

The process of named entity recognition based on CRF is shown in figure 4. Firstly, the original detection data is divided into training set A and test set B, and then corpus is generated. After the crf-based model training, the first named entity recognition and information extraction was carried out by crf_test function.

---

**Figure 4.** Named entity recognition based on CRF  
CRF is a probabilistic structural model used to segment and annotate sequence data. Given the observed sequence X, the conditional probability P(Y|X) of the output annotated sequence Y is calculated. Compared with other sequential annotation models such as HMM and Maximum Entropy Markov Mode (MEMM), CRF weakens the independence hypothesis. CRF only needs to consider the characteristics of observed sequences that have already appeared. It can make full use of context information, and it is easy to fuse different features. Meanwhile, it can carry out parameter optimization and decoding in the global scope, avoiding Label Bias problems that may occur in MEMM and other discriminant Markov models. The key difference between CRF and MEMM is that MEMM uses an exponential model of each state to determine the conditional probability of the next state of the current state. CRF, on the other hand, uses a single exponential model to calculate the joint probability of the entire annotation sequence and a given observation sequence. Therefore, the weights of different features in different states can be substituted with each other. CRF can be thought of as an undirected graph model or Markov random field. Theoretically, the graph structure can be arbitrary as long as some conditional independence is expressed in the annotation sequence. However, the most simple and common first-order Chain structure is generally used to solve the problem of sequence annotation, as shown in figure 5.
Figure 5. The first order chain structure of CRF

We define $X = x_1, x_2, \cdots, x_n$ as the given observation sequence, that is, the corpus of software monitoring data composed of n words. $Y = y_1, y_2, \cdots, y_n$ is the annotation sequence of the output, that is, the predicted entity annotation sequence. The conditional probability of the output sequence can be defined as equation (1).

$$P(Y \mid X, \lambda) = \frac{1}{Z(X)} \exp \left( \sum_j \lambda_j f_j(y_{i-1}, y_i, X, i) + \sum_k \mu_k s_k(y_i, X, i) \right)$$  \hspace{1cm} (1)

$Z(X)$ is the Normalization Factor, which can make the sum of all possible state sequence probabilities equal to 1 and it can be obtained by equation (2). $f_j(y_{i-1}, y_i, X, i) \}$ is the transition function, which represents the transition probability marked on the current position $i$ and previous position $i-1$ of the observation sequence $X$. $s_k(y_i, X, i)$ is a state function, representing the annotation probability of the current position $i$. The above two functions are collectively referred to as eigenfunctions and both depend on local features. In the process of named entity recognition, when the feature template condition is met, the value is 1. Otherwise, the value is 0. $\lambda_j$ and $\mu_k$ are the corresponding weights of $f_j$ and $s_k$ respectively, which can be estimated on the model training set by the Maximum likelihood function.

$$Z(X) = \sum_y \exp \left( \sum_j \lambda_j f_j(y_{i-1}, y_i, X, i) + \sum_k \mu_k s_k(y_i, X, i) \right)$$  \hspace{1cm} (2)

After the eigenfunction weight is obtained, the training process of the model is basically completed. The observation sequence $X$ is input into this model, and the named entity annotation sequence $Y$ with the highest probability can be decoded by Oiterbi algorithm, as shown in equation (3).

$$Y' = \arg\max P(Y \mid X, \lambda)$$  \hspace{1cm} (3)

3.3.2. Named entity recognition and extraction based on Rule-based method

After the CRF-based named entity recognition, through the analysis of the error cases in the output results, we can make artificial rules to further improve the recognition performance by analyzing the error situation in the output result. The process of second named entities recognition and information extraction based on Rule-based method is shown in figure 6.
3.4. Overall structure
The experimental environment adopts Python3.6. After starting the running environment, the main components are loaded as follows:

- `import re`: String match
- `import Jieba`: Word segmentation toolkit
- `import nltk`: Natural Language Toolkit
- `import sklearn`: Data preprocessing toolkit

The main modules used are as follows:

- `Jieba.cut`
- `Jieba.cut_for_search`
- `nltk.stem`
- `nltk.tokenize`
- `sklearn.feature_extraction.text`

4. Conclusion
This paper presents a scheme for processing software monitoring data based on NLP. Firstly, the data are preprocessed by word segmentation and data cleaning. Then the named entity recognition and information extraction are accomplished by fusing CRF and rule-based methods. The scheme is suitable for software monitoring data processing under any operating system. By characterizing the software monitoring data with the method proposed in this paper, it can provide effective data support for system fault prediction and make data preparation for fault prediction and prevention.
References

[1] Machiraju, Srikanth and R. Modi. 2008 Natural Language Processing Developing Bots with Microsoft Bots Framework

[2] Grishman R. 2019 Twenty-five years of information extraction. Natural Language Engineering 1-16.

[3] Raitio T, Suni A, Yamagishi J, et al 2011 HMM-Based Speech Synthesis Utilizing Glottal Inverse Filtering IEEE Transactions on Audio Speech & Language Processing 19(1) 153-65

[4] Lin Yao, Chengjie Sun, Shaofeng Li, Xiaolong Wang and Xuan Wang 2009 CRF-based Active Learning for Chinese Named Entity Recognition. IEEE International Conference on Systems. IEEE.