A Code Classification Method Based on TF-IDF

Ke WANG*, Jian-Hong JIANGa, Rui-Yun MAb

Business College, Guilin University of Electronic Technology, Guilin Guangxi, 541004, China
*1821840368@qq.com, ajjhome@guet.edu.cn, b799427060@qq.com

Keywords: Code classification; Code duplication; Clustering Algorithm; TF-IDF.

Abstract. The main purpose of the study is to find the code with similar possibilities to effectively avoid the adverse effects of code duplication. Through the clustering pretreatment of document feature information, to extract the relevant features of the document. Then the basic characteristics are used to cluster the document, to find out the best number of clusters. According to the reasonable number of clusters that have been found, using the vectors that generated through TF-IDF method, combined the K-means clustering algorithm to distinguish the contents of the files, as well as the introduction of cosine similarity, to determine the similarity of two texts and classify the parallel documents. From the test data set, the method can accurately find the code with the possibility of duplication and works quiet well.

Introduction

At present, with the progress of computer network technology and continuous development of network collaboration sharing mode, program design and development has become very convenient. Developers can find the implementation of corresponding code only through searching the network for related program logic and requirements, which also provides a very handy condition for network plagiarism. However, at the same time, copy of the code will also cause the impact of possible bug becomes larger, consequently, there is a great demand of distinguishing analogical code or identical text efficiently.

From the research of domestic and abroad, code similarity detection mainly focused on two aspects: text similarity detection and program logic similarity detection. The study of text comparability includes the comparison of texts and the comparison of words. Many scholars utilize TF-IDF to establish textual features. For instance, Trstenjak B et al.[1] combined kNN method to do text classification, Zhuo K et al.[2] applied the TF-IDF method to test the similitude of papers. Chen K et al.[3] employed the improved TF-IDF: TF-IGM to determine the weight of the word and partition the text. There are also some scholars Luo Q et al.[4] by means of the semantic word weight method for text assortment and recognition. Chen Y-T et al.[5] classified text by integrate chi-square statistics with cosine similarity method. Jiang C[6] take keyword as the basic unit, by considering the semantic similarity and the combination order between keywords, to calculate the similarity of texts from the perspective of word meaning and word order of keywords. Wu Q et al.[7] put forward the method of keywords overlap, using single keyword as semantic unit to measure the similarity of texts.

Another study[8]based on procedural logic, including the formation of an abstract syntax tree (AST) after the program was compiled to detect the comparability, or the detection of the program dependency graph (PDG). Zhuo Li et al.[9] combined the dynamic text matching algorithm with suffix tree algorithm for similitude code within source files, achieved a similar code detection tool, actually united the method of abstract syntax tree. Michel Chilowicz et al.[10] through the factorization of the function call graphs, detected the similarity of source code from the function level. Sharma A et al.[11,12] determined the similarity of two functions according to the similarity of the internal operating instructions, and eventually get the similarity of the two applications.
Type Identification of Code Text

The practical situation in the study is: most of the documents are in a folder or a subfolder inside the folder, the number of documents is very large, direct comparison will consume superabundant system resources and time. In order to detect the repetition rate of the document, a preliminary clustering must be first made to identify the approximate type of text. And the determination of the cluster number need to be resolved by a certain test algorithm. Based above, more accurate features are applied to detect the similarity of the document. Therefore, the primary task of the study is to preprocess the text and extract the relevant features of the document. Then the basic characteristics are used to cluster the document, to acquire the best number of clusters. On the basis of the cluster number, the information of TFIDF as well as the improved K-means algorithm is combined to distinguish the text. Based on the above process, the calculation framework of this paper is as follows:

Figure 1. Code Classification Calculation Framework.

Text Preprocessing and Eigenvalue Acquisition

The distinct from ordinary document is that there are fewer language-related issues in the code text, for example, there is no need to address some stop words and altofrequency punctuation. Code document can get two types of indicators: text storage features and text phrase features. The former is a quick access to acquire information, the latter requires a certain amount of time to calculate. The text discrimination information for this study is calculated from these two types of indicators.

**Fundamental Characteristics of the Text.** For the document class code, through the text statistical analysis can obtain some elementary features that the document contains.

- Number of words in the document: The number of all the words contained in the text.
- Number of non-repetitive words in the document: Remove the number of duplicate words from all the words in the text.
- Number of characters in the document: Read text by characters, then counts the number of character contained in the text.

**Word Phrase Feature of the Text.** By TF-IDF (Term Frequency-Inverse Document Frequency), the main idea of TF-IDF is: If the frequency TF of a word or phrase appeared in an article is high, and rarely appeared in other articles, it is considered that the word or phrase has a good classification ability, suitable for classification. TF-IDF is a statistical method, applied to evaluate the importance of a word to a set of files or one of the documents in a corpus. The importance of the word is proportional
increase to the number of times it appeared in the document, but simultaneously, will be inversely proportional decrease to the frequency it appeared in the corpus. The high term frequency of a particular document, and the low document frequency of the term in the entire file collection, can produce a high weight TF-IDF. Accordingly, TF-IDF tends to filter out common words and retain significant words [13].

Term frequency (TF) refers to the occurrence frequency of a given word in the document. That is the ratio between the times of word \( w \) appeared in document \( d \) Count(\( w, d \)) and the total number of words size(\( d \)) in document \( d \).

\[
   tf(t, d) = \frac{f_{t, d}}{\sum_{i \in d} f_{i, d}}
\]

Inverse Document Frequency (IDF) is a measure of the general importance of words, the IDF of a given word, can be expressed as the total number of files divided the number of files which containing the term. And then take the logarithm from the obtained entropy. That is, the logarithm of the ratio between the total number of documents \( N \) and the number of documents \( w \) docs(\( W, D \)) appeared.

\[
   idf(t, D) = \log \frac{N}{|d \in D: t \in d|}
\]

Where \( N \) is the total number of documents for the corpus | \( D \) |, according to \( tf \) and \( idf \), calculate a weight for each document \( d \) and the query string \( q \) composed by keyword \( w \) [1] ... \( w \) [\( k \)], to indicate the matching degree between query string \( q \) and document \( d \).

\[
   tfidf(t, d, D) = tf(t, d) \times idf(t, D)
\]

**Type Classification of Document Content**

Vectors that generated through TF-IDF method, combined the K-means clustering algorithm to distinguish the contents of the file, the basic idea of K-means algorithm is: Initially, assign K cluster centers randomly, divide the sample points which need to be classified into each cluster according to the nearest neighbor principle. And then recalculate the center of each cluster in accordance with the average method, to determine the new cluster center. Iterating until the moving distance of the cluster center is less than the given one. And the distance of the computing cluster centroid is determined by calculating the resemblance between the documents, in this survey, cosine similarity will be applied in the vector space model, to determine the semblance between two texts [14, 15].

Assuming that \( A \) and \( B \) are two n-dimensional vectors, and the similarity calculation formula for \( A \) and \( B \) is as follows:

\[
   \cos \theta = \frac{\sum_{i=1}^{n} (A_i \times B_i)}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}
\]

The algorithm are as follows:

Figure 2. Cluster File Label.
1) Given a data set which contains multiple data points;
2) Select K centroids randomly;
3) Calculate the data points in the data set respectively, determine which group of centroids the data points belong to, and gather all data points in the data set into K groups;
4) According to the K groups of data points calculated last step, to recalculate a new centroid respectively;
5) Repeat step 3, and carry out a clustering process again, to get K groups of data points;
6) Calculate the new centroid again, if the distance change between the new calculated centroid and the last calculated centroid is very small (meet the prescribed threshold, or convergence), consider the results are in line with expectations, and stop the clustering process.

By using the algorithm the above shows to calculate the test data, gained the corresponding label for each of the following files:

Finally, according to the classification results, find the most similar top 10 groups of data from the test data set, the data are as follows, the left string is the file name, the middle number is the identified category, the right number indicates the number of occurrences for the same file name which belongs to the category.

((3-1.php', 1), 26)
((3-4.php', 3), 18)
((math.php', 9), 16)
((3-1br.php', 1), 16)
((3-3.php', 0), 14)
((3-2.php', 5), 13)
((math.php', 4), 12)
((3-3.php', 7), 10)
((math.php', 2), 9)
((3-4.php', 6), 7)

Conclusion

From the final processing results, the effect of calculation processing is good, by comparing the same file name, the top 10 groups of similar documents accounted for all files of 78.3%. Further study found that the file name is not alike, but the content is analogous, after the amendment, the first 10 groups of parallel files accounted for 83.8% of all documents. After finding the analogical file, by calculating the cosine similarity as well as presenting a threshold, to find the most likely repeat file set.

This study conducted the clustering pretreatment by using document feature information. According to the reasonable number of clusters that have been found, using the improved k-means combined tf-idf algorithm to classify the parallel documents, from the test data set, the method can accurately find the code with the possibility of duplication and works quiet well.

Acknowledgement

This research was financially supported by Guangxi National Science Foundation under Grant No.2014GXNSFGA118287 and Guangxi Education Department General Project under Grant KY2015YB114 and Ministry of Education Humanities and Social Sciences Fund Youth Project under Grant 17YJHZH074.

References


