Research of Distinct Algorithm of Short Text Based on Simhash

Yun ZHANG\textsuperscript{1,2}, Zong-ze JIN\textsuperscript{1,2}, Wei-min MU\textsuperscript{1,2} and Wei-ping WANG\textsuperscript{2}

\textsuperscript{1}School of Cyber Security, University of Chinese Academy of Sciences, China
\textsuperscript{2}Institute of Information Engineering, CAS, China

Keywords: Simhash, Text de-duplication, Data clean.

Abstract. With the development of social network, microblog is the typical application in big data era. However, there are two aspects: the one is Chinese language is more various and flexible than English, the other is that microblog with Simhash is not good. This paper analyzes the data of microblog, which is the big number of data and short text, then the innovational algorithm is introduced. We used Simhash as the baseline, then we improved it and proposed B-Simhash algorithm. Meanwhile, we focused on the data quality of microblog by special character processing. Through experiments we can gain the results of B-Simhash, which is better in processing the short text. Adding the special character processing, the result is better than the original. According to the result of experiments, the higher precision and recall rate are gained. The efficiency is also improved for microblog de-duplication.

Introduction

With the development of the big data, social networks play an important role in people’s daily life, such as Twitter [1], FaceBook [2,3], etc. The microblogs of sina and tencent are also very popular in China. Due to the frequent Communication of users in the social networks, hot topics and hot events spread quickly. At the same time, due to the low threshold of the news data release of the social networks and the relevant imperfect laws, it is very necessary to detect the sensitive information.

The analysis of online crime network ecosystem has been researched by Sinha [4] et al. They tried to use social relationships and semantic information to find more junk accounts based on some seeds account of the computer crime. Maclean [5] believed that the opening transmission was the main reason for more and more garbage data. He pointed out that the opening potential topic transmission should be limited for reducing the garbage data.

According to the different principles of the algorithm, the text de-duplication algorithm is divided into two categories [6,7]. The SCAM [8,9] based on vector space model was established by Shivakumar and Garcia-Molina. And then the graph structure was used to store text test [10].

The above researches are carried out around the English version Twitter. As far as the Chinese text de-duplication technology is concerned, it mainly uses the existing methods of English text de-duplication technology. And it is mainly applied to the Chinese Web page de-duplication. There is little research on microblogs in China. Chinese language is more various and flexible, therefore the study of microblog is more difficult. This paper focuses on the detection of the Chinese junk microblog based on Simhash algorithm. It can do better than most approaches about detecting garbage data in social network. The experimental results show it is effective and reliable.

Related Work

Simhash

Simhash algorithm [4,8] has been proposed by Charikar in Google. It converts two text records into two n-bit fingerprint. And then the similarity can be computed based on the two n-bit fingerprint. If the two n-bit fingerprint is closed, the two text records are similar.

From the generation to the comparison of the two fingerprint, the process is not involved in text contents. Therefore, Simhash has been widely applied to distinct the massive similar text. It tackles the problem, which is not necessary to store massive data in big data scene.
The specific process of the Simhash, as shown in figure 1:

The testing text will be split, and get n word segmentation.

An n-dimensional vector will be initialized, and each bit of the vector is given the initial value of 0; An F-bit hash fingerprint of each word segmentations will be gained by computing. Then, the fingerprint will be overlying. And it creates the f-bit final fingerprint. The final fingerprint is the Simhash fingerprint of the text. (Google recommends f=64).

The two Simhash fingerprints will be compared for computing the hamming distance. When the hamming distance is less than a threshold value, it is similar. (Google recommends the threshold value equals 3).

**Bloom Filter**

Bloom Filter [2,10] has been proposed by Burton Howard Bloom in 1970. Bloom Filter is a kind of random data structure with a high spatial efficiency. Firstly, it creates a bit array and gives each bit the initial value of 0. Each element in the testing sets is calculated by k hash functions. Each hash function maps the element to k points in a bit array and sets the k points the value 1. When a new element comes into the testing set, it also is calculated by k hash functions above. The value of each points of the new element check in the bit array, then the similarity will be gained.

The process is as follows:
1. If the value of any of these points of the new element equals 0, it is not the existing element.
2. If the value of each points equals 1, it is maybe the existing element.

The advantage of Bloom Filter is that the space efficiency and the query time are better than other algorithms.

However, if an element appears in the testing set, the element will go through the Bloom filter. At the same time, the element is not in the testing set, it is also possible to go through the bloom filter. Therefore, the disadvantage of Bloom Filter is false positive.

**B-Simhash**

**Problem Description**

Bloom filter and Simhash are used widely in the field of the traditional text de-duplication, but there are defects between them.

BloomFilter is based on memory, so it needs a better processing capacity. However, the defects of the algorithm are also obvious. Because of its high false positive rate, there are many duplicate text records in the result of de-duplication.

Simhash algorithm not only can distinct the texts, but also can improve the accuracy of checking similar text by comparing the Hamming distance. However, the complexity of the Simhash algorithm is larger than the Bloom Filter algorithm. Therefore, the efficiency of Simhash algorithm is lower than Bloom Filter.

In view of the advantages and disadvantages of them, the B-Simhash algorithm is proposed in this paper. B-Simhash algorithm is based on Bloom Filter and Simhash. First, the process of pretreatment
can be effective to filter most of the duplicate text through the Bloom Filter. We use Simhash algorithm to deal with the results after preprocessing. Finally, the results are more accurate.

The Process of B-Simhash

According to the Figure 2, the process is as follows:

The testing text is calculated through Bloom Filter. The eight random number generators (hash function \( f_1, f_2, \ldots, f_8 \)) produced the eight fingerprints information for the testing text. And then the bit position of the eight fingerprints information is gained.

If the fingerprint is found in the testing set, it means that the text is a duplicate text. Otherwise, the text is not.

If the testing text is through the Bloom Filter, it is analyzed by Simhash algorithm. The testing text will be splitted to some segments. The weight of each segment will be calculated, and the hash value is gained. According to the hash value, Simhash fingerprint is calculated. The Simhash fingerprint of the testing text is compared with the Simhash fingerprint of the testing set. Hamming distance will be calculated. If Hamming distance is less than the threshold, the testing text is a duplicate text.

The following details of the B-Simhash algorithm shows as:

It assumes that the testing text is \( y \) and the testing set is \( S(x_1, x_2, \ldots, x_n) \). An \( m \) dimensional array \( A \) is established in memory and each bit in \( A \), it sets 0.

A set of 8 hash functions \( (H_1, H_2 \ldots H_8) \) is gained. The text can be mapped into the \( A \) by any of the 8 hash functions. Therefore, the database of fingerprints can be established. Every element of the \( S \) will be mapped into the fingerprint database by any of the 8 hash functions and the corresponding bit will be set 1.

There are 8 results of the testing text \( y \) through the 8 hash functions.

\[
(H_1(y), H_2(y) \ldots H_8(y))
\]  \( (1) \)

The \( (H_1(y), H_2(y) \ldots H_8(y)) \) is compared with the corresponding bit of \( A \). We can determine whether the corresponding bit is 1.

When the above comparison is passed (the corresponding bit is 1), the testing text is a duplicate text. If the above comparison is not passed (Any of the corresponding bit is not 1), the testing text is not a duplicate text.

If the testing text \( y \) is not repeated, we should calculate the similarity. The testing text is divided into \( k \) segments \( (y_1, y_2 \ldots y_k) \) through the word segmentation function \( U \). Then every segment is
calculated by the weight function \( V \). The segment-weight \( [(y_1, w_1), (y_2, w_2) \ldots (y_k, w_k)] \) is established, and it will be set into the Set \( \mathcal{C} \).

\[
\mathcal{C} = V(y_1, y_2 \ldots y_k) = [(y_1, w_1), (y_2, w_2) \ldots (y_k, w_k)]
\]  

(2)

Every element of \( \mathcal{C} \) is calculated by hash functions \( H \). The 64-bit hash code-weight is established. A set \( T \) of 64-bit hash code-weight is established.

\[
T = [(H(y_1), w_1), (H(y_2), w_2) \ldots (H(y_k), w_k)]
\]  

(3)

We can calculate the Simhash fingerprint of \( T \). We can get every fingerprint of segments with weights. And then, every hash code will be superimposed. If the result is more than 1, it sets 1, otherwise it sets 0. This is the Simhash fingerprint of \( y \).

\[
\text{Simhash fingerprint} = \sum_{i=0}^{k} (H(y_i) \ast w_i)
\]  

(4)

We can calculate the Simhash fingerprint of \( y \) is compared with each of \( S \) to calculate the Hamming distance. If the Hamming distance is less than the threshold, it is a similar text, otherwise it is not.

Through the above method, the duplicated or similar text in the microblog can be found with a high accuracy and recall rate.

**Data Clean**

In the process of samples detection, some problems are found in the microblog content. Details are as follows:

1. Too much junk information exists, which is incomplete, such as “comment”, “link”, “nightly-night” and so on.
2. Too many foreign characters exist, especially like Japanese kanas. They are usually accounted as junk information, such as “No.946dklbr ㄏ か ょ ョ ㄈ ㄈ ぜ と ヾ シ っ ケ カ ヰ ん カ け No.342sjteh てづはじァ゜ゎㄕ”.
3. The condition of key keywords appears, they are often discerned as junk information like “weather forecast”, “horoscope of today” and so on.

There are plenty of meaningless words in the process of first segmentation for word, such as “not only”, “some” and so on. They affect the accuracy and reduce the processing efficiency.

For the three conditions above, we design three kinds of algorithms to deal with these problems in this paper as following:

1. Filtering the text length. We set the threshold value of length to ten, in other words, if the length of measured text is less to ten, it is filtered as the junk information.
2. Filtering Japanese kanas. If the text includes Japanese kanas, it will be filtered.
3. Setting the corpus of keywords. When the segment word and stop word are matched with the one that has been in the corpus, it will be filtered.
4. Filtering stop word. We can set stop word corpus, and we use 1-level segment word except the stop word to analyze.

In the first stage of data accessing, we use these clean rules to enhance our data quality, meanwhile, it can improve our algorithms capacities.

**Experiment**

**Dataset**

According to the crawling interface provided by sina, the microblog content is from August 2015 to October 2015. By the extraction of data respectively, it gets three groups of datasets as follows.

1. Group1: We extract 9 million random microblog content from the dataset in August 2015, meanwhile, and we simulate 1 million similar text, there is 10 million blog content as the test dataset.
(2) Group2: We extract 9 million random microblog content from the dataset in September 2015, meanwhile, and we simulate 1 million similar text, there is 10 million microblog content as the test dataset.

(3) Group3: We extract 9 million random microblog content from the dataset in October 2015, meanwhile, and we simulate 1 million similar text, there is 10 million microblog content as the test dataset of this experiment.

Evaluation

We adopt precision and recall rate as the evaluation in our experiments.

The precision shows as:

\[ P = \frac{TP}{TP + FP} \]  

(5)
P is the precision, TP is the correct number of samples, FP is the incorrect number of samples.

The recall rate shows as:

\[ R = \frac{TP}{TP + FN} \]  

(6)
R is the recall rate, TP is the correct number of samples, FN is the number of samples which have not be detected.

Details and Analysis

(1) B-Simhash, Original Simhash and Bloom Filter

We adopt three groups of datasets above. The experiment shows the comparative result with the B-Simhash Algorithm, the original Simhash Algorithm and Bloom Filter.

In the results, the computing time, precision and recall rate show as:

From the figure 3, the B-Simhash shows better efficiency than the Original Simhash. However, the efficiency of Bloom Filter is highest in the three algorithm.

From the figure 4 and 5, B-Simhash algorithm shows certain promotion both in precision and recall rate certain, compared with Bloom Filter and original Simhash.

(2) Data cleaning set and the original set
From the figure 6 and 7, the result shows that the better precision and recall rate are gained with the date cleaning set.

(3) Analysis

According to the result of the experiments, B-Simhash algorithm has a higher precision and recall rate. At the same time, it is a better efficiency. To a certain extent it solves the problem of bloom filter (false positive).

Because the first step of B-Simhash is running in memory and it reduces the data size for the second step, B-Simhash is a better efficiency than the original simhash algorithm.

Beyond that, according to the actual data situation of microblog, the special characters’ filter is applied to clean data. The data set is cleaner for follow algorithm. Therefore, the precision and recall rate are further improved.

However, according to the result, there are still some problem. B-Simhash has a lower precision and recall rate for original simhash although it has a better efficiency. We will develop the follow deeper research work for above problem.

Conclusion

We combine with the scene of massive microblogs in processing with cleaning and de-duplicated. Meanwhile, we should solve the problem that we should pay attention to take measures to process the special characters. In this paper, we use Simhash algorithm as the baseline, meanwhile, we consider the execution details which can influence our results to optimize. Through repeated experiments, we propose B-Simhash algorithm. We use it to process our real scene to tackle the problem which is the massive garbage data in microblogs. As the results, we gain the higher accuracy and recall rate than other algorithms.

Acknowledgment

This paper is supported by grant 61502478 from Research on key technologies of short text sentiment analysis.

References


