Study on Similarity Compute and File Filtering Based on Cloud Computing Method

Ma yuanyuan[1] Zhang bo[2] Wang yufei[3]
[1]China electric power research institute
[1]mayuanyuan@epri.sgcc.com.cn [2]zhangbo3@epri.sgcc.com.cn [3]wangyufei3@epri.sgcc.com.cn

Abstract-Text similarity computing has been widely used in confidential document filtering to enhance the safety of an enterprise information system. And the accuracy rate and performance of the similarity computing has always been the crucial problem in the research of document filtering. With the approaching era of massive data, the traditional way of computing similarity can not meet the needs of enterprises any more, but new ideas can be put forward in cloud computing environment. Aiming to solve this problem, this paper presents an algorithm of computing the distributed similarity which is based on mutual information document in cloud computing environment. This algorithm can calculate the text similarity based on cloud computing environment, and the calculations can be used to achieve the document filtering function. We've lanuched some experiments in Hadoop cloud computing environment, and the results show that this algorithm is a high-performance and effective algorithm.

Keywords: Text Filtering; Cloud Computing; Text similarity.

1. Introduction

As the strong development of Internet, it brings splendid burst of information, which forces us to face a tough test on how to process enormous data. To solve the problem of "Abundant data, slum information", it requires us to analyze and explorer the immersing data. When it becomes difficult to screen available information among the generous ocean or to screen out confidential information, the concept of "Text Filtering" emerges[1].Text filtering refers to a process enable special users to get qualified text from a huge mess of texts[2]. Given a certain requirement by the user and a stream of entry text, the Text Filtering system is able to judge whether each text in the stream fits the demand of user according to the established initial profile of users. Then it will inform the user of qualified texts, and then modify the user pattern in accordance with the result judged by the user on texts screened out in favor of the demand of users [3].

Relevance manifested among the texts screened out through compute on similarities is an portant process in Text Filtering [4]. The process of Text Category is to classify an unknown text in terms of its content through a given categorizing method, which can be either classified as different categories or into none categories (refer to the categories given) [5]. The procedures to categorize texts are: word segmentation; choosing typical words; texts redefining; categorization confirming. The four procedures above are linked, that is to say, each one preceding is to be entered in the following procedure [6].

During the third procedure, one of the ways to present the text is VSM (Vector Space Model), which means to take the texts as multi-dimensional vectors, presenting their interrelation with cosine similarities. However, vectors are of high dimensions. For overcoming this major disadvantage, it requires the system to extract feature words for reducing excessive vectors. A method called Mutual Information is used for figuring out interrelation between a word and a certain category. In another word, it measures the mutuality of two objects. Mutual Information is one of the concepts provided by Information Theory, referring to the relation between information that represents statistical correlation of two random variables with quantitative values. Feature extraction through Mutual Information Theory is based on a hypothesis: if a lexical item appears high frequency in a certain category, but lower in other categories that means it carries high mutual information with the former. Mutual information between a feature item and a category demonstrates the degree of correlation of the two factors. This standard is widely used to establish statistic model of words correlation. To sum up, in a certain subject, the bigger the mutual information of a feature is, the higher its probability of subject concurrence will be. Therefore, mutual information can be used as the most evident feature during an evaluation process of feature extraction.

As it mentioned above, VSM (Vector Space Model) is a way to represent the texts, taking a text as a multi-dimension vector, by which the correlation between texts can be turned into that between vectors. In this way, cosine law can be used to work out the similarity between texts. It is a usual way to categorize texts based on their similarities. However, the most significant drawback is the high dimension of vectors. To reduce it, feature items of texts are considered having particular importance. The most complicated stage of text categorization is to extract feature items, as well as the presentation of texts, especially during the categorization of enormous data.

An effective way to settle large data computing is called Distributing Cloud Computing, distributing huge data which to be computed into different computers, and then bringing the results together. It is an instant way for large data computing. Hadoop is a typical environment of Cloud Computing, settling memory, analysis and compute on large data sets. In recent period, it is applied to many platforms for its advantages of low cost, flexibility, efficiency, high fault tolerance, etc. Major components of Cloud Computing include Hadoop Distributing File System (HDFS) and MapReduce programming model. The main ideas of MapReduce are to divide the task and collect the results. The function of MapReduce can be attributed into two verbs, namely Map and Reduce. Map means to divide a task into different subtasks. On the contrary, Reduce is used to collect the results of the divided, eventually into a combination.

In current, many mature computing on file categorization are on the basis of single tasks. As long as combining the extraction of featuring words and presentation procedures of files with the distributing computing framework, it will significantly increases the categorization rate and the number of corpuses, and will speed up text similarity computing and efficiency of text filtering.

2. Relevant Tasks

The 4 current methods of filtering content are as follows: PCIS-based filtering (Platform for Internet Content Selection), URL-based filtering (Uniform Resource Location), ill-meaning keywords and intellectual-concept-based filtering. PCIS-based filtering is performed by a user or an administrator through security settings of Explorer to filter content on webpages. The method requires some rating labels with additional annotation attached by the issuing party of a webpage when it is initially publicized. For example, CIRA (www.fosi.org/cira) can rate information automatically in accordance with that annotation. As a result, the PCIS-based filtering can only be used for assistance. The URL-based filtering performing through the setting of black or white lists keeps a high efficiency and a speedy rate, which is easy to be put into practice, but with less flexibility, laying high difficulty on management.

Major technologies applied in content filtering include: similarity of texts, semantic comprehension and model matching algorithm.

The development of pattern string match technology relates highly to its application. The technological matching of pattern string is applied to build a data FTR system (Full Text Retrieval System) and a system for enquiring book contents and abstracts. In recent years, with the ever development of Internet technology and biotechnology, profound interests are aroused in Pattern String Matching Technology (PSMT). However, the huger number of data unmatched poses a new challenge to PSMT. In fact, it has become a bottleneck in Information Filter System (IFS) and other systems like Intrusion Detection System (IDS). Therefore, a more applicable matching algorithm has been becoming a focus for research. This technology is extending single pattern matching algorithm to multi-pattern matching as well as analyzing classic algorithms, such as AC, WM, FS and Compressed Encoding algorithms.

The statistics-based method of text similarity measurement takes a text as a collection of individual words, modeling the text as high-dimension vectors but few and scattered. It figures out inter-textual similarity with cosine similarities of vectors or Jaccard similarity. This method covers VSM, LSI (Latent Semantic Indexing) model, a method based on Attribute Theory, etc. Advantages of the method above are convenience in computing and comprehension and its wide usage. Nevertheless, its disadvantages include: requirement of large-scale corpus, negligence of semantic relation inter-textual and high-dimension model but few and scattered, resulting in difficulties in processing. The most significant method of textual similarity computing in statistics processing is text categorization. The mission of automatic text categorization is to process texts voluntarily and decide if the predefining category targets one or more categories. As the number of electronic-format texts increase at an exponential speed, it becomes growingly important and difficult to index effective information, conduct content management and filter information.

The similarity computing method based on semantic analysis aims at building semantic correlation between words through knowledge base in special areas, for the purpose of reviewing text similarity. In comparison with statistics-based computing method, this one depends on large-scale corpuses, but has a high accuracy. However, to build a knowledge base is a complicated task. Current studies employ complete dictionaries instead of knowledge base. Compared with the statistics-based methods for similarity computing, the one based on sematic comprehension need neither large-scale corpuses nor long-time training for support. It is highly accurate, using WordNet for similarity computing such as disambiguation researches “Agirre” etc; using synonym collection for similarity computing, such as researches on computing similarity between sentences; using structure of Hownet knowledge to conduct similarity computing, such as words or semantic similarity researches.

In current, the main stream algorithms on text categorization are working on single computer, instead of the distribution characteristics of text. In addition, many text processing methods in the presenting stage calculate the weigh value of lexical categorization through computing word frequency. However, the methods above lead to some deviations. This thesis focuses on distributed texts with mutual information, combining VSM for computing of text similarity in order to achieve the goal of categorization.

3. Similarity Computing Method Based on Mutual Text Presentation

In the process of huge text filtering, text filtering performance can be straight affected by bottleneck problems such as scale restriction of data process, shortage of performance and accuracy in similarity computing. The thesis proposes a similarity-computing method based on distributed mutual texts, including the following procedures: to collect and initialize the target texts, to compute the frequency of automatic words and the mutual information value in different categories as well as sorting out a collection of feature words, to compute the weigh value of all the feature words to form an eventual collection of vectors and conduct a final similarity computing based on VSM. On one hand, the rate and extendibility of text categorization can be improved by MapReduce, a distributing computing framework for extracting feature words and weigh value. On the other hand, through the design of key-value, the weigh value of feature words in the text can be figured out by parallel computing when extracting featuring words, improving the rate and accuracy of similarity computing as well.

The definition $D=\{d_1, d_2,..., d_j,..., d_N\}$ means corpus where d_j refers to each file of the corpus and N refers to the number of files in the corpus. $F=\{f_1, f_2,..., f_k,..., f_{|F|}\}$ means the collection of feature words where $|F|$ refers to the total number of the feature words and $0<k\leq|F|$, k belongs to integers, additionally, f_k refers to each feature words. $C=\{c_1, c_2,..., c_s,..., c_{|C|}\}$ means a categorized collection of files, therein c_s refers to the category of each file and $|C|$ refers to the number of file categories. $TF=\{tf_{11},..., tf_{ij},..., tf_{MN}\}$ means a frequency

collection of automatic words, therein i represents serial numbers of automatic words, j refers to the serial number of a file， M represents the number of different word categories, and N represents the number of all files in the corpus, the elements in this collection refer the frequency of automatic word t_i in d_j .

All the files in the corpus must be converted into files easy to access, then conduct segmentation on those files. The results include the contents and types of word segmentations. The files processed will be taken as input splits of MapReduce distributed into different DataNodes, in favor of the convenience of processing.

To simplify the segmentation process of those input splits, it has to remove pause words, punctuations and single-word segmentation. Simplification process is implemented as follow: to process Task A whose input splits are processed files where each file represents an independent input splice of data. The key-value pair entry of Task A in Map stage is <<filename，texttype>，context>, therein, “filename” refers to the name of a file, “texttype” refers to the name of its category, and “context” refers to the content of the file. Key-value pair output of Task A in Map stage is <<filename，texttype，word>，1>，therein, “word” refers to segmentation of each files; “1” refers to the time it occurs. This key-value pair is the entry of Task A in Reduce stage.

To calculate the word frequency with MapReduce is to compute the mutual-information value of segmentations with 3 consecutive sub-MapReduces. They are respectively defined as A, B and C.

In the Reduce stage of Task A, to count the same filenames and key-value pares will get the frequency of segmentations in each file. <<filename，texttype，word>，wordcount> is the result of Task A output, therein, “wordcount” refers to segmentation frequency in each files. Then the result of Task A must be saved in TF , the collection of segmentation frequency. Then Task C has to be established, calculating $\alpha+\beta$. The eventual output of Task B is taken as the entry of Task C where Map stage takes an individual segmentation and its filename as the key value. When α appears, it must be taken count as one time, namely <word，<filename，texttype， α ， 1>>, then integrates all of them into simplification task during the simplification stage to calculate the times a segmentation appears in all the files in all categories, namely $\alpha+\beta$. As a result, the final output of Task C is <<word， texttype>， < α ， $\alpha+\beta$ >>.

In Task D, output in Task C helps to calculate $MI(t_i c_j)$. In Map stage, computation with key-value entry in accordance with formula (1), then get an output, <<word， texttype>， $MI(t_i c_j)$ >. In Reduce stage, key-value pairs with of the same file category shall be entered into the same simplification task, sorting as $MI(t_i c_j)$ in Reduce stage and picking out the high mutual-information segmentations as the eventual feature words, namely <texttype，word> . Finally, save it into F , collection of feature words.

The definition vector $\overrightarrow{W_j}=\{w_{1j},w_{2j}...w_{kj}...w_{|F|j}\}$ represents the weight vectors of all feature words, $0<k\le|F|$, k belongs to integers; w_{kj} represents feature words in file d_j ; t_i refers to segmentations in file d_j . w_{kj} is represented as:

$$w_{kj}=\left\{\frac{tf_{ij}}{\sum_{j=0}^{N}tf_{ij}}|tf_{ij}\in TF,\ f_k\in F,\ t_i=f_k,0\le i<M\right\}$$

Definition collection is the collection of all file vectors, which is the final presentation of files.

The collection above is the vector entry of similarity algorithm based on VSM in favor of similarity computing between texts.

4. Performance Test

We implement the algorithm above through a computing environment built by Hadoop, which based on MapReduce, a distributing computing model. The platform includes 5 data nodes and 1 namenode. We upload all the files onto the data nodes in an average manner. Before starting the algorithm, all files must be processed for segmentation (using word segmentation algorithm of Chinese Academy of Science), removing related pausing words, as the entry of the algorithm. The ICTCLAS segmentation system of CAS is adopted in this system. ICTCLAS 2011 has a segmentation rate of 500KB/s and an accuracy of 98.45%, with an API no more than 100KB and a compacted dictionary data no more than 3M. To draw a comparison on performance parameters, we studied 800 files and 200 files to be tested and computed their similarities of threshold value.

We use another two similarity algorithms for comparison. The first one is based on Jaccard parameter of aggregation model, which is called Jaccard for short. This method computes text segmentations and weight values, and present text similarity algorithm with Jaccard parameter. The second method is based on word similarity of semantic comprehension, which is called Semanteme for short. This algorithm combines the generalized VSM with word similarity of semantic comprehension to calculate the ultimate similarities between texts.

The table below is the results of those three methods. The one provided by this thesis is called DSM for short. We can draw a conclusion that the DSM can achieve a better result when meeting a relatively large threshold value, and less time cost.

Category	DSM method	Jaccard method	Semanteme method
Time of performance (ms)	98666	116812	183536
Threshold value of similarity	0.061000	0.007210	0.053000
Acceptability	44.3%	50.2%	39.7%

Configure1. Performance comparison of different algorithms

5. Conclusion

This thesis provides a similarity algorithm based on mutual-information texts for effective text categorizing. It computes weight values of texts with mutual information, combining with the similarity computing method based on VSM. It uses a distributing model, for improving the efficiency of huge text filtering. The thesis verifies the feasibility and performance of the method through related experiment results.

Although the similarity algorithm proposed in this thesis can improve the accuracy of text filtering, there exists a problem that the method is an off-line process, when the text base changes, we have to learn it again. This problem makes the application of the text filtering process difficult to deal with. One of our focuses in the coming period is to optimize its performance for the realization of on-line text categorization (stream processing model), coordinating different MapReduce tasks to get a further improvement in text filtering.

References

[1] GUO Qing-lin, LIYan-mei, TANG Qi “The calculation of documents similarity based on VSM”, Application Research of Computer, 2008, 25(11):3256-3258.

[2] GAO Jie, JI Gen-lin “Survey onText CategorizationTechniques” Application Research of Computer, 2004, (7):28-31.

[3] Spiros Papaimitriou, J M Sun. DisCo: Distributed Co-clustering with Map-Reduce.Proc of the 2008 Eighth IEEE International Conference on Data Mining. Washington DC,IEEE Computer Society, 2008: 521-521.

[4] Chao Liu, Hung-chih Yang, Jinliang Fan, et al. Distributed nonnegative matrix factorization for web-scale dyadic data analysis on mapreduce, International Conference on WWW. New York, ACM, 2010: 681-690.

[5]Zhang Jia-yong,Hu Jian-hui.Intelligent information filtering systems based on Chinese word segmentation Information Technology,2006:175-178.

[6] G Mann, R McDonald, M Mohri, N Silberman, and D Walker. Efficient large-scale distributed training of conditional maximum entropy models. Advances in NeuralInformation Processing Systems 22, 2009: 1231–1239.