Smart Cloud Document Clustering and Plagiarism Checker using TF-IDF Based on Cosine Similarity

Sudhir Sahani
Student
Department of Computer Science and Engineering
IMSEC Ghaziabad, India

Rajat Goyal
Student
Department of Computer Science and Engineering
IMSEC Ghaziabad, India

Saurabh Sharma
Student
Department of Computer Science and Engineering
IMSEC Ghaziabad, India

Shaili Gupta
Assistant Professor
Department of Computer Science and Engineering
IMSEC Ghaziabad, India

Abstract

This research paper describes the results oriented from experimental study of conventional document clustering techniques implemented in the commercial spaces so far. Particularly, we compared main approaches related to document clustering, agglomerative hierarchical document clustering and K-means. Though this paper, we generates and implement checker's algorithms which deals with the duplicacy of the document content with the rest of the documents in the cloud. We also generate algorithm required to deals with the classification of the cloud data. The classification in this algorithm is done on the basis of the date of data uploaded and. We will take the ratio of both vectors and generate a score which rates the document in the classification.
Keywords- Algorithm, Cloud, Classification, Hierarchical, Clustering

I. INTRODUCTION

Most of the devices now-a-days used the cloud platform to store their data. The devices are running on the cloud. Many of the applications are running on the cloud. Cloud contains a lot of identical data in it. While user tries to upload a document to cloud, he may counterfeit if the document uploaded are found to be plagiarized. On the other hand files are classified on the basis of k-means document clustering algorithms to make better provisions. But sometimes clustering may not achieved. This will affect the efficiency of the cloud. The network cost is also high while accessing the cloud because the data is not transmitted from your neighbourhood, it is from miles of distance from a server on which it resides. This miles of distance may also effect the efficiency in accessing the cloud service. There are a lot more errors in accessing the data at a distance server. These all are the issues in today's cloud service.

We considering "TEXT" type of data to make the cloud to do smarter operations on the server side and make the services more efficient. For that we use to process the document when the user wants it to upload it to the cloud, before uploading, the server process the content of the document and matches the content with the uploaded content in cloud, if there is more than X%(value depends on user) of content matches with the previously uploaded document, the cloud generates a warning to the user. It depends on the server, what type of restrictions it provides to user to upload identical document or not. As the document content not matched with the previous uploads, then document is uploaded on the cloud. This process is used to not allow the duplicate content in the cloud. The second operation is to classify the data in such a manner that user finds it desired document on cloud. The classification of the documents depends on both the scenarios, further, there are algorithms to describe how to proceed these operations of cloud and makes the smart cloud.

II. IDEA BEHIND SMART CLOUD

The idea behind developing the concept is to moderate the duplicacy in the online document clustering. Suppose a user wants to publish a document file or review paper, this system will evaluates and check the pre-existing documents with the new document in order to achieve plagiarized free content. Thus it help in ensuring plagiarized free document.

Considering the scenario of college, suppose a student wants to publish a review paper over any journalism website. Then his document will be reviewed by the system before final upload. It will undergoes several algorithmic processes. In most cases it helps show the % of duplicate content if encountered. It consequently help in overcoming the problem of original document. Thus reducing the proportion of copied content. Several algorithm like tf-idf, k-means clustering, and cosine algorithms makes the difference in achieving this concept.

III. OVERVIEW

A. TF-IDF [1][2][3][4]

TF-IDF is generally a content descriptive mechanism for the documents. The term frequency (TF) is the number of times a term appears in a document and is calculated as follows: tf = (Number of occurrences of the keyword in that particular document) / (Total number of keywords in the document). Inverse Document Frequency (IDF) measures the rarity of a term in the whole corpus. Denoting the total number of documents in a collection by N, the inverse document frequency of a term t is defined as follows: idf = log (N / df). The concepts of term frequency and inverse document frequency [11] are combined, to produce a composite weight for each term in each document. tf-idf = tf * idf.

Table 1: Recommended TF-IDF weighting schemes

Weighting Scheme	*Document term weight*	*Query term weight*
1	$F_{t,d}$. $logN/n_t$	$(0.5+0.5f_{t,q}/max_t f_{t,q}).logN/n_t$
2	$1 + logf_{t,d}$	$log(1+N/n_t)$
3	$(1+logf_{t,d}).logN/n_t$	$(1+logf_{t,q}).logN/n_t$

B. Cosine Similarity Measure [5] [6]

There are many techniques to measure the similarity between the user query and the retrieved documents. One of such widely used technique is cosine-similarity [11]. It is one of the powerful similarity checking technique compare to all the other techniques exist [14] and widely used for web

Document similarity. Cosine-similarity (q,d) = (2)

Where, q and d are query and document vectors respectively. Also ||q|| and ||d|| represent their length respectively. The strength of the similarity depends on the value of θ. If θ = 00, then the document and query vector are similar. As θ changes from 00 to 900, the similarity between the document and query decreases.

The cosine of two non-zero vectors can be derived by using the Euclidean dot product formula:

$$a.b = ||a|| b|| cos\Theta$$

C. Tokenization [7]

Tokenization is a process of parsing data text into smaller units called tokens usually termed as phrases or words. It uses Bag-of-words Model and N-gram model. Initially the documents are complete pack of phrases with bundles of stopping keywords. It widely helps it eliminating the white spaces and punctuation. This method is fairly straightforward in case of those uses inter word spaces.

D. Stemming [8]

It is a process of reducing the derived words to their stem word. This consequently helps in reducing the overhead from the plagiarization checker. It is widely used by search engines and treat words with same stem as synonyms, this is termed as conflation. In general simple stemmer looks up the inflected form in a lookup table. This approach is very agile and easily handles exceptions. Sometimes it follows Suffix Stripping algorithm because it doesn't depend on lookup table.

if the word ends in 'ed', remove the 'ed'
if the word ends in 'ing', remove the 'ing'.
if the word ends in 'ly', remove the 'ly'.

It is approximate method for grouping similar basic word together and involves stochastic algorithms which uses probabilistic approach to identify the root form of a word.

E. Stop Words []

Those words which are non-descriptive in nature termed as stopping words. During document clustering such kind of words are eliminated to avoid exceptions and thereby helping faster any group of words can be chosen as the stop words for a given purpose. The most common stopping words are 'the, are, is, at, on, take, that, etc.

IV. RELATED WORK

A. Software-Assisted Detection [9]

In terms of software Assisted detection, the plagiarism detection has been straightforward in recent times. With more dependency onto digitized documents, it has been highly practiced. One of the highly used method is In Text Document where system works in two generic detection technique one is intrinsic other being external.

External detection systems compare a suspicious document with a reference collection, which is a set of documents assumed to be genuine.

B. String Matching [10]

It is one of crucial method used in computer science. In this documents are compared for verbatim text overlaps. Checking a suspicious document in this setting requires high the computation and storage of efficiently comparable representations for all documents in the reference collection to compare them pairwise. In this suffix vectors or suffix trees are used for achieving the task.

V. CONCLUSION

It will make the system more robust and allow users to publish documents plagiarized free. With the inclusion of more peculiar and better algorithmic approach, it enables faster result. It will block the user if the new document contains plagiarized content as that of pre-existing document over the cloud.

ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the template and our faculty members whom immense support during the project development.

REFERENCES

[1] Rajaraman, A.; Ullman, J. D. (2011). "Data Mining". Mining of Massive Datasets (PDF). pp. 1–17. doi:10.1017/CBO9781139058452.002. ISBN 9781139058452.
[2] Beel, Joeran; Breitinger, Corinna (2017). "Evaluating the CC-IDF citation-weighting scheme - How effectively can 'Inverse Document Frequency' (IDF) be applied to references?" (PDF). Proceedings of the 12th iConference."
[3] Robertson, S. (2004). "Understanding inverse document frequency: On theoretical arguments for IDF". Journal of Documentation. 60 (5): 503–520. doi:10.1108/00220410410560582Valtchev, Stanimir S.; Baikova, Elena N.; Jorge, Luis R. (December 2012). "Electromagnetic Field as the Wireless Transporter of Energy" (PDF). *Facta*Universitatis Ser. Electrical Engineering (Serbia: University of Niš) 25 (3): 171–181. doi:10.2298/FUEE1203171V
[4] Manning, C. D.; Raghavan, P.; Schutze, H. (2008). "Scoring, term weighting, and the vector space model". Introduction to Information Retrieval (PDF). p. 100. doi:10.1017/CBO9780511809071.007. ISBN 9780511809071.Leyh, G. E.; Kennan, M. D. (September 28, 2008). Efficient wireless transmission of power using resonators with coupled electric fields (PDF). NAPS 2008 40th North American Power Symposium, Calgary, September 28–30, 2008. Inst. of Electrical and Electronic Engineers. pp. 1–4. doi:10.1109/NAPS.2008.5307364.
[5] Singhal, Amit (2001). "Modern Information Retrieval: A Brief Overview". Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24 (4): 35–43./
[6] Graham L. Giller (2012). "The Statistical Properties of Random Bitstreams and the Sampling Distribution of Cosine Similarity". Giller Investments Research Notes (20121024/1). doi:10.2139/ssrn.2167044.
[7] Huang, C., Simon, P., Hsieh, S., & Prevot, L. (2007) Rethinking Chinese Word Segmentation: Tokenization, Character Classification, or Word break Identification
[8] Lovins, Julie Beth (1968). "Development of a Stemming Algorithm". Mechanical Translation and Computational Linguistics. 11: 22–31.
[9] https://en.wikipedia.org/wiki/Plagiarism_detection
[10] Baker, Brenda S. (February 1993), On Finding Duplication in Strings and Software (gs) (Technical Report), AT&T Bell Laboratories, NJ