Hashing Technique Based on Duplication Detection in News Articles

Ms. Nupur Gaikwad
PG student,
Computer Engineering Department,
Alamuri Ratnamala College of Engineering,
Mumbai

Prof. Sanjay Jadhav
Associate Professor,
Dept. of Computer Engineering,
MGM College of Engineering and Technology,
Mumbai

***Abstract*:- Now a days as we know electronic media has been developing very fast, resulting in a large number of news articles produced online, and thus duplication detection is needed. Besides article duplication is directly related to the to articles plagiarism. Existing studies in news articles duplication detection maily focuses on news articles , and we further explore duplication detection in news articles from the newest online We Media data. In this paper , we propose the tool , NDFinder , using fingerprinting techniques with hash index to detect article duplication.**

Keywords: Duplication Detection, News Article, Plagrrism Detection, Big Data .SSS

INTRODUCTION

Duplication detection is very useful for a variety of tasks (e.g., file management, copyright protection and plagiarism prevention). However, existing works about duplication detection mainly focus on documents or code duplication detection, and only a few works aim at duplication detection in news articles. Moreover, existing works just detect and analyze duplication in newspaper articles [2]. Previously, only press agencies could publish news articles, but now the advent of We Media makes everyone can publish and share news online. There lacks duplication detection and analysis based on the newest We Media data. Hence, we focus on the study of duplication detection in news articles from We Media which contain larger number of data.

In this paper, in order to support accurate duplication detection of news articles, we propose a technique and implement it as a tool, called NDFinder. We first normalize the content of each news article, such as removing spaces and images. Then we design a hash-based algorithm to process each sentence in the article, which can be seen as a fingerprinting technique. In other words, each news article is processed as a set of hash values, where each hash value represents a text sentence in the article. Furthermore, for pairs of obtained hash set (i.e., pairs of articles), NDFinder uses Jaccard similarity coefficient as similarity function to detect duplication.

Motivation

Meanwhile, a large number of news articles are produced online, and articles duplication can no longer be ignored, since duplicate articles will increase the redundancy and management costs. Moreover, articles duplication is directly related to articles plagiarism. In the field of journalism, plagiarism is considered a breach of journalistic ethics, so it is very important to detect duplication in news articles.

Hashing Algorithm

Fingerprinting algorithm is used, and signatures (i.e., hash values) are calculated per line in the content of article. This approach is sensitive to minor modifications made in duplicate content. Assume A={a1,a2,...,an} is a set of normalized text data, where an represents the content of n-th article. Then, since each article can be seen as the set of sentences, for each article's each sentence, a hash value v is computed.

Jaccard similarity Algorithm:

Jaccard Similarity is a common proximity measurement used to compute the similarity between two objects, such as two text documents. Jaccard similarity can be used to find the similarity between two asymmetric binary vectors or to find the similarity between two sets. In literature, Jaccard similarity, symbolized by J, can also be referred to as Jaccard Index, Jaccard Coefficient, Jaccard Dissimilarity, and Jaccard Distance.

Jaccard Similarity is frequently used in data science applications. Example use cases for Jaccard Similarity:
Text mining: find the similarity between two text documents using the number of terms used in both documents

Methodology

The entire process of our proposed approach is summarized in Fig. 1. It can be considered in two phases: (1) Normalization; and (2) Detection. The following subsections describe the detailed design of each phase.

A. Normalization

In the normalization processing phase, key components of each article are first extracted from the news articles data, which are organized as the characteristic matrix: <Title, Abstract, Content, AuthorId, AuthorName, PublicTime, CommentNum, Category, Link>, where each element is a characteristic vector containing corresponding information of all articles. Although we aim at the duplication detection of Content in news articles, other components of the articles are

also important, such as Title, AuthorName and PublicTime, since such information can further guide the empirical analysis on news articles plagiarism.
We focus on the duplication detection of articles content, and thus we further process and normalize the content of each article. Based on the observation, we find that plagiarists often changed the position of images in the article to avoid plagiarism detection.

B. Detection

Given a set of normalized text data, we use fingerprinting algorithm combining with hash technique to process each article's data. We evaluate pairs of articles' text are similar by measuring their ratio of matched hash values (i.e., matched sentences), and report those satisfying the similarity threshold.

1) Fingerprinting and Hashing: In this step, fingerprinting algorithm is used, and signatures (i.e., hash values) are calculated per line in the content of article. This approach is sensitive to minor modifications made in duplicate content.

2) Measuring and Reporting: This step is to verify which articles are duplicate, and report the duplicate articles. For each article pair in set A, we check whether they are duplicate by measuring the similarity θ of their corresponding hash set .

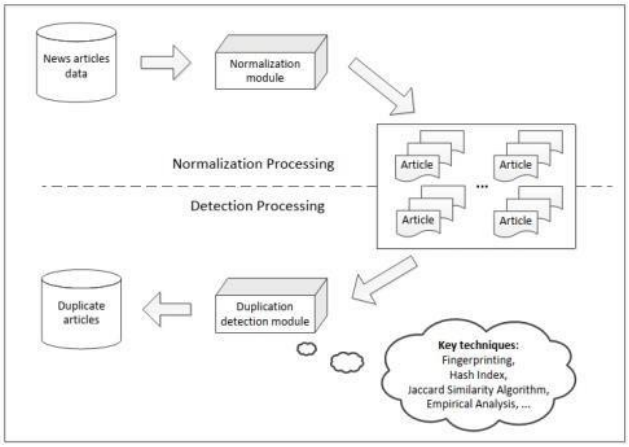

Fig : Architecture Diagram

Data Design

A description of all data structures including internal, global, and temporary data structures, database design (tables), file formats. Internal software data structure Dataset news articles are an internal data structure. Global data structure Trained features data which are available to major portions of the architecture are global data structure. Temporary data structure Articles words extracted are interim or temporary data structure.

Database

To validate our proposed approach, we first crawled big data of news articles. Then we evaluate our tool NDFinder by detecting duplication in these collected news articles. Since we aim at detecting duplication in news articles from We Media, we use Toutiao, one of the most popular and famous We Media in China, as our target. We randomly crawled news articles from Toutiao's site by a bunch of web crawlers. According to different news channels, we divide these collected data into seven categories: Game, History, Military, Sports, Technology, Entertainment and Food.

Algorithm 1: Duplication Detection

Input: A is a set of normalized text data $\{a_1, a_2, ..., a_n\}$, where each element represents one article's content. θ is the similarity threshold, which can be specified by the user.
Output: Pairs of duplicate articles

$hashSubSet = \phi$;
$hashSet = \phi$;
for $i = 1 \rightarrow n$ **do**
  for *each sentence* s *in* a_i **do**
    $v = Hash(s)$;
    $hashSubSet.insert(v)$;
  end
  $hashSet(a_i).insert(hashSubSet)$;
  $hashSubSet = \phi$;
end
$dupPair = \phi$;
for $j = 1 \rightarrow n$ **do**
  for $k = j + 1 \rightarrow n$ **do**
    setIntersection$(hashSet_{a_j}, hashSet_{a_k})$;
    setUnion$(hashSet_{a_j}, hashSet_{a_k})$;
    $num1$ = num of hash value in Intersection Set;
    $num2$ = num of hash value in Union Set;
    if $num1/num2 >= \theta$ **then**
      $dupPair = dupPair \cup (a_j, a_k)$;
    end
  end
end
return $dupPair$;

CONCLUSION

We have presented an article duplication detecting technique, and implemented as a tool, NDFinder. We find that the top three topics with the highest proportion of duplication are Sports news, Military news, and Technology news.

REFERENCE:

[1] Carter, D., Stojanovic, M., and de Bruijn, B. (2018). Revitalizing the Global Public Health Intelligence Network (GPHIN). Online Journal of Public Health Informatics, 10(1).
[2] Chatterjee, S. (2019). Rage within the machine: Brexit headline blizzard overloads FX algos. Reuters, Apr
[3] Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In International conference on machine learning, pages 957–966.
[4] Arun, P. and Sumesh, M. (2015). Near-duplicate web page detection by enhanced TDW and simHash technique. In 2015 International Conference on Computing and Network Communications (CoCoNet), pages 765– 770. IEEE.
[5] Software Engineering and Service Sciences, pages 495–499.IEEE
Mathew, M., Das, S. N., and Vijayaraghavan, P. K. (2011). A novel approach for near-duplicate detection of web pages using TDW matrix. International Journal of Computer Applications, 19(7):16–21.
[6] Ling, Y., Tao, X., and Lv, H. (2010). A priority-based

method of near-duplicated text information of web pages deletion. In 2010 IEEE International Conference on

[7] Hajishirzi, H., Yih, W.-t., and Kolcz, A. (2010). Adaptive near-duplicate detection via similarity learning. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pages 419–426. ACM.

[8] Blei, D. M. and Lafferty, J. D. (2009). Topic models. In Text mining, pages 101–124. Chapman and Hall/CRC. Broder, A. Z., Glassman, S. C., Manasse, M. S., and Zweig,
G. (1997). Syntactic clustering of the web.Computer Networks and ISDN Systems, 29(8-13):1157–1166.

[9] Gibson, J., Wellner, B., and Lubar, S. (2008). Identification of duplicate news stories in web pages. In Proceedings of the 4th Web as Corpus Workshop

[10] Henzinger, M. (2006). Finding near-duplicate web pages: a large-scale evaluation of algorithms. In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 284–291. ACM.

[11] Broder, A. Z. (2000). Identifying and filtering nearduplicate documents. In Annual Symposium on Combinatorial Pattern Matching, pages 1–10. Springer.

[12] Chowdhury, A., Frieder, O., Grossman, D., and McCabe M. C. (2002). Collection statistics for fast duplicate document detection. ACM Transactions on Information Systems (TOIS), 20(2):171–191.

[13] Broder, A. Z. (1997). On the resemblance and containment of documents. In Compression and complexity of sequences 1997. proceedings, pages 21–29. IEEE.

[14] Broder, A. Z. (2000). Identifying and filtering nearduplicate documents. In Annual Symposium on Combinatorial Pattern Matching, pages 1–10. Springer.

[15] L. Lu and P. Wang. Duplication and plagiarism detection results.[Online].Available: http://home.ustc.edu.cn/%7Ewpc520/data.rar.