ATLANTIS PRESS

Advances in Engineering Research, volume 138

2nd International Conference on Machinery, Electronics and Control Simulation (MECS 2017)

General Simhash-based Framework for News Aggregators

Pengcheng Hu[1,a] and Xiangdong You[1,b,*]

[1]Beijing University of Posts and Telecommunications, Haidian, Beijing, China

[a] hupengcheng@bupt.edu.cn, [b] youxiangdong@bupt.edu.cn

*corresponding author

Keywords: News Aggregator, Simhash, Deduplication, News Recommendation, Breaking Event Detection.

Abstract: News aggregator usually indexes billions of news from Internet and try to recommend news according to readers' intrinsic interests. Retrieval for similar news, deduplication and event detection are common problems in aggregator systems, and related works are reported in [1], [2], [3], [4] and [5]. We proposed a general simhash-based framework for news aggregator, the system has no necessary to process crawled news for retrieval, deduplication and event detection respectively, each piece of news is processed only one time and without extra storage space. Duplicates and breaking events can be detected online before new crawled news was stored in system's database. Machine learning are widely used in news aggregator for tasks like topic classification and each piece of news is mapped into a feature vector with fixed length. Simhash fingerprints are generated on feature vectors rather than original text of news, therefore news retrieval, deduplication and breaking news detection can be integrated into any running aggregator systems without extra efforts. Our aggregator collected around 9.6 million of news from Internet and the framework function well in real scenario.

1. Introduction

An emerging trend in modern press industry is providing digit news to readers. Users can browse news on a website belongs to a certain news service provider, however, the website just show the same contents to all users regardless of their interests and browsing history. Meanwhile, users need to browse different websites if he subscribes several news services.

A news aggregator is a system which aggregates online newspapers from different sources and provide personalized recommendation to users. The system need to crawl thousands of news daily from different news sources and clean them into well-formatted text. Then the system also need to calculate similarity between any two news so that it can recommend similar news to users according to their browsing history. A news aggregator can also provide special reports which contains series of news of a continuous event from different news providers. At the same time, news aggregator is an automatic system of news recommendation, it should be capable of detecting breaking events like natural disasters or political scandals, when no editors involved in.

Deduplication is crucial when news aggregator leverages web crawler to collect online news. Web crawler, also called spider, is an Internet robot that systematically browses the Internet to collect information. It usually starts with a list of URLs to visit and crawls all contents of these website pages. It will identify all hyperlinks in the page and add them to the list of URLs to visit, and repeat the process until there is no URLs in the list. Perhaps there are multiple pages contain the same hyperlink which link to the same webpage and it's wasteful to visit the same URLs several times. It's easy for us to handle this problem by using a cache table like a Bloom filter, however, it's hard for us to tell whether two different websites have highly similar contents, and we called it near-duplicates. It's common that one news provider will reproduce the same news from another provider. Obviously, it's essential for news aggregator system to identify near-duplicative news.

It's also crucial to recommend similar or related news to users when they have finished browsing a piece of news. An implicit assumption of user's browsing behaviour is that users tend to read

news of similar topics. It's reasonable to recommend similar news to users according to the news he is browsing.

News aggregator relies heavily on automatic algorithms and it's essential for aggregator systems to be capable of detecting breaking events in time when there is no human editor involved in. News aggregator should also provide special reports of continuous event which can be collected from different news providers.

Most aggregator systems will provide category labels for each news like Politics, Entertainment and Sports, etc. The system usually mapping text of each news into a fix-length vector and training a multi-class classifier which use these vector as features. Similarity can also be calculated with cosine distance of two feature vectors. However, a big challenge of news aggregator system is the issue of scale: the system will index billions of news in the database and crawl thousands of news per day.

In this paper, we proposed a general framework to handle deduplication, retrieval for similar news and breaking events detection very quickly in billions of news.

2. Framework of news aggregator

Our framework consists of several independent parts. Web crawler keeps crawling news from different sources. After a piece of news was crawled from Internet, feature extractor will map the text of news into a fix-length feature vector. Simhash generator is responsible for generating a 64 bits fingerprint for feature vector of news. Deduplication and breaking events detection will be processed before storing the new crawled news into database. The new crawled news won't be stored in our database if duplicates are detected. And breaking events will be collected together to form a special report if they are detected.

2.1. Feature vector representation of news

The news aggregator system need to map the text of each piece of news into a fix-length vector before further processing. There are many different techniques for feature extraction, however, BoW models [6] and distributed representation [7] is the most popular methods used in NLP. TF-IDF [8] is one of the widely used method of BoW models, it's simplistic but surprisingly useful in practice. Another dominated method is distributed representation which map text into highly compressed vectors, doc2vec [9] for example.

Preprocessing is necessary regardless of which representation the system will adopt, TF-IDF or doc2vec. Preprocessing usually contains several sequential steps including tokenization [10], remove stop words [11] and stemming [12]. Tokenization is various for different languages and particularly difficult for languages which exhibit no word boundaries such as Ancient Greeks and Chinese. Stop words means that the frequency of these words is too high and we can benefit little from these words, so we can just remove these words from text. Stemming is the process of reducing derived words to their word stem, base or root form. For grammatical reasons, two semantic similar sentences may use different forms of word, it would be useful to turn these words into a common base form.

We also need to reduce feature dimension after feature extraction, because high dimension feature will do damage to performance. Dimensionality reduction [13] is the process of reducing the number of feature vector's dimension. It can be divided into feature selection [14] and feature transformation. Feature selection approaches try to find a subset of the original features according to the weights of each feature. However, feature transformation usually tends to keep each feature by mapping the feature vector from a high-dimensional space to a low-dimensional space. PCA [15] (principal component analysis) and LDA [16] (linear discriminant analysis) are common transformation techniques.

2.2. Fingerprints Generation and Retrieval

Each piece of news is represented by a fix-length feature vector after feature extraction. Similarity between two news can be calculated by cosine distance of their vectors, however, it's

hard for us to find the top N news with highest similarity when specific news provided. The time complexity of similar news retrieval grows rapidly when the size of the indexed news continuously increases. We can calculate the similarity of each pair of news, but the time cost is too huge to accept. By leveraging simhash, we can find most similar news in constant time in a database which contains billions of news.

2.2.1. What is Simhash?

Simhash was proposed by Charikar in [17], it's a dimension reduction technique which maps high-dimensional vectors to small-sized fingerprints. In section 2.1, we have discussed how to extract feature vectors for each piece of news and these vectors usually have high dimension more than 30k. We can use simhash to map these vectors to a f-bit fingerprint where f is small, say 64 or 128. One import property of simhash is: Similar feature vectors will have similar fingerprints. If two fingerprints are similar, the Hamming distance of these two fingerprints is small, says d. We can retrieve similar news using news' fingerprints.

2.2.2. Fingerprints Generation

Supposing dimension of feature vector extracted from text of news is d, we can generate f-bit fingerprint as follows:

We initialize an array V of length f , each element of V is set to 0. For each feature in the feature vector, we use a hash function to hash the feature into a f-bit hash value. We use the f-bit hash value to update the array V: if the i-th bit of the hash value is 1, we increase the i-th element of V by 1; if the i-th bit of the hash value is 0, we decrease the i-th element of V by 1. After we processed all features in the feature vector, values of elements in V are positive or negative. We generate the f-bit fingerprint F according to the sign of elements in V: set the i-th bit of F to 1 if the i-th element of V is positive, or the i-th bit of F will be set to 0.

2.2.3. Fingerprints Retrieval

Now that we have generated a f-bit fingerprint for each feature vector, how do we quickly retrieve other fingerprints that are similar with the given one? Similar fingerprints generated by simhash means that the Hamming distance of these fingerprints are small, says d. How do we quickly retrieve fingerprints that differ in d bit-positions?

Suppose we have 10 million of news stored in the aggregator system and the fingerprints is 64-bit. The number of fingerprints is closed to 2^{23} if the system generates one fingerprint for each piece of news. We can also assume that Hamming distance of two fingerprints, F and F', should be small than a threshold, says d, if two news are semantic similar. We can conclude that |f-d| bit-positions are same for F and F', although we have no idea the exact d-positions where two fingerprints are different. If f-bits is splited into N blocks, where N > d, then there must be at least one block that all bits are same for F and F'.

By leveraging this finding, we can store f-bit fingerprint into N buckets and each bucket is associated with one of N blocks. Two fingerprints will be stored in the same bucket if they have identical block. We can retrieve similar fingerprints just in N buckets rather than all buckets. Let's consider the following 3 different designs:

1. Split 64 bits into 8 blocks, each contains 8 bits. If we have 2^{23} fingerprints, then there will be 2^{23-8} = 32768 fingerprints in each bucket. On average, a probe retrieves 8 * 32768 = 262144 fingerprints.
2. Split 64 bits into 4 blocks, each contains 16 bits. There will be 2^{23-16} = 128 fingerprints in each bucket. On average, a probe retrieves 4 * 128 = 512 fingerprints.
3. Splits 64 bits into 5 blocks having 13, 13, 13, 13 and 12 bits respectively. There will be 2^{23-13} = 1024 fingerprints in each bucket and a probe will retrieve 5120 fingerprints on average.

Although we have nearly 10 million of news in database, a probe retrieves only 5 thousand of news, the probe space is reduced significantly. By retrieving fingerprints in this way, the system can find similar news in constant time even billions of news are stored in database, as long as size of f and storage plan is well-designed.

2.3. Deduplication, Similar News Retrieval and Breaking Event Detection

We have discussed generation and retrieval of simhash fingerprints, then how do we use this technique to handle deduplication, similar news retrieval and breaking events detection?

Manku et al. [18] states that Hamming distance of near-duplicates is close to 3. And we also found that two news are similar rather than near-duplicative if the Hamming distance is bigger than 3 but small than 8.

2.3.1. Deduplication and Similar News Retrieval

We will retrieve similar fingerprints whenever a new piece of news is crawled. If there are fingerprints that differs only in 3 or less bit-positions with the new crawled news, we will not persistent the new crawled news into our database due to near-duplicates.

When we want to recommend similar news to readers, we will retrieve similar fingerprints and find out fingerprints whose Hamming distance is small than 8 and bigger than 3 with the provided one. Then we will recommend corresponding news to readers, and we can rank news according to the similarity with the currently reading one.

2.3.2. Breaking Event Detection

Breaking events, like natural disasters or political scandals, usually happened unexpectedly. Breaking events will be paid great attention and the public want to know and discuss all details of these events. Different news providers will report the event and huge amount of news will be reported in a short time. If the aggregator system found that lots of similar news was crawled in last few hours, we can assume that the system has detected breaking event. The system can collect all related news into a special report and recommend it to all readers.

2.4. Experimental Results

Our news aggregator frameworks heavily rely on simhash to detect similarity. We first dive into relationship between Hamming distance of simhash fingerprints and similarity of news. Next, we analysed performance of event detection of our system.

In our experiments, we used Blob Service of Microsoft Azure to store simhash fingerprints. Blob Service is a NonSQL database which can persistent non-structured data. It consists of blobs and each blob is identified by a key. A blob is just like a bucket, you can store any object in a blob, files or images for example, as long as stored object has a unique key in this blob.

For each fingerprint, we split f bits into N blocks, and each block is the key to one Azure blob that the fingerprint will be stored in. A fingerprint is stored in N blobs and each retrieval also need to probe these N blobs (Figure 1). We want to analyse retrieval time for one fingerprint so that we can estimate performance of our system. Retrieval time for fingerprint F was defined as: For each fingerprint F' in the same blob that F was stored, we calculate Hamming distance between F and F', and time cost for all fingerprints will be summed up.

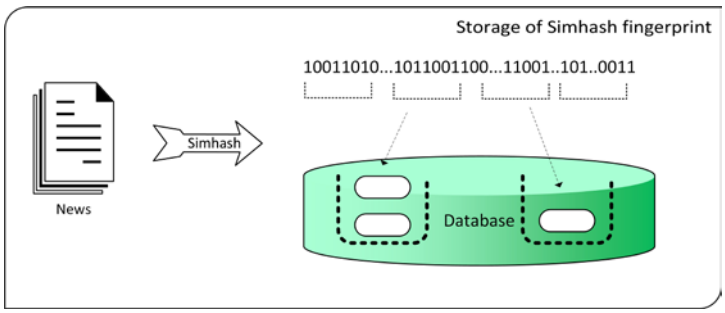

Figure 1 Generation and storage of simhash fingerprints.

2.4.1. Fingerprints Storage

Our news aggregator has nearly 9.6 million of news crawled from Internet and we generated a 64 bits fingerprint for each piece of news. In Section 2.2.3, we proposed three different storage plans

and analysed the theoretical probe space of each plan. Our experiments test the retrieval time of plan 2 and plan 3, when a given fingerprint provided.

We randomly sampled 10000 news and corresponding fingerprints from our database. Table 1 shows average retrieval time and precision of plan 2 and plan 3:

Table 1 Comparison of block length: 16 and 13.

Block bits	Retrieval time
13,13,13,13,12	0.019747 s
16,16,16,16	0.003646 s

2.4.2. Breaking Event Detection

We select 20 big events reviewed by editors from our recommendation history. There are more than 30 news in each event and a big event was detected only if: For fingerprint F, more than 15 similar news was retrieved and all of them are crawled within 4 hours, then a big event was detected.

3. Conclusions

In this paper, we proposed a general news aggregator framework which leverage simhash to handle deduplication, similar news retrieval and breaking events detection. By generating simhash fingerprints on the feature vector of news rather than original text, any aggregator can integrate these functionalities into it's running system without extra efforts to alter existing framework. Although aggregators have different scales, well-designed storage plan for fingerprints can guarantee constant time of news retrieval.

References

[1] Lee, M. D., Navarro, D. J., & Nikkerud, H. (2005, January). An empirical evaluation of models of text document similarity. In Proceedings of the Cognitive Science Society (Vol. 27, No. 27).

[2] Elsayed, T., Lin, J., & Oard, D. W. (2008, June). Pairwise document similarity in large collections with MapReduce. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers (pp. 265-268). Association for Computational Linguistics.

[3] Lillibridge, M., Eshghi, K., Bhagwat, D., Deolalikar, V., Trezis, G., & Camble, P. (2009, February). Sparse Indexing: Large Scale, Inline Deduplication Using Sampling and Locality. In Fast (Vol. 9, pp. 111-123).

[4] Sakaki, T., Okazaki, M., & Matsuo, Y. (2010, April). Earthquake shakes Twitter users: real-time event detection by social sensors. In Proceedings of the 19th international conference on World wide web (pp. 851-860). ACM.

[5] Allan, J., Papka, R., & Lavrenko, V. (1998, August). On-line new event detection and tracking. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 37-45). ACM.

[6] Zhang, Y., Jin, R., & Zhou, Z. H. (2010). Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics, 1(1-4), 43-52.

[7] Hinton, G. E. (1984). Distributed representations.

[8] Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. Information Processing & Management, 39(1), 45-65.

[9] Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (ICML-14) (pp. 1188-1196).

[10] Mcnamee, P., & Mayfield, J. (2004). Character n-gram tokenization for European language text retrieval. Information retrieval, 7(1), 73-97.

[11] Wilbur, W. J., & Sirotkin, K. (1992). The automatic identification of stop words. Journal of information science, 18(1), 45-55.

[12] Paice, C. D. (1994, August). An evaluation method for stemming algorithms. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 42-50). Springer-Verlag New York, Inc.

[13] Fodor, I. K. (2002). A survey of dimension reduction techniques. Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 9, 1-18.

[14] Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

[15] Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52.

[16] Prince, S. J., & Elder, J. H. (2007, October). Probabilistic linear discriminant analysis for inferences about identity. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on (pp. 1-8). IEEE.

[17] Charikar, M. S. (2002, May). Similarity estimation techniques from rounding algorithms. In Proceedings of the thiry-fourth annual ACM symposium on Theory of computing (pp. 380-388). ACM.

[18] Manku, G. S., Jain, A., & Das Sarma, A. (2007, May). Detecting near-duplicates for web crawling. In Proceedings of the 16th international conference on World Wide Web (pp. 141-150). ACM.