Fake and Real news detection

Christodoulos Constantinides[1], Giannis Alexandrou[2] and Kristian Litsis[3]

[1] Department of Computer Science, University of Cyprus, Nicosia, Cyprus
[2] Department of Computer Science, University of Cyprus, Nicosia, Cyprus
[3] Department of Computer Science, University of Cyprus, Nicosia, Cyprus

E-mail: ccons04@cs.ucy.ac.cy
E-mail: galexa05@cs.ucy.ac.cy
E-mail: lkrist01@cs.ucy.ac.cy

Received xxxxxx
Accepted for publication xxxxxx
Published xxxxxx

Abstract

"Fake news" was not a term many people used four years ago,but nowadays it is arguably one of the most serious challenges facing the news and media industry today. Fake news is written and published usually with the intent to mislead in order to damage an agency, entity, or person, and/or gain financially or politically advantages,[3][4][5] often using sensationalist, dishonest, or outright fabricated headlines to increase readership. Similarly, clickbait stories and headlines earn advertising revenue from this activity.[3] Fake news undermines serious media coverage and makes it more difficult for journalists to cover significant news stories.[6]

General Terms Algorithms

Keywords Hamming distance, near-duplicate, similarity, search, sketch, fingerprint, web crawl, web document

---

1. Introduction

Duplication of content on the Internet is inevitable, whether it be intentional or accidental, malicious or coincidental. There are several types of duplicate documents.True duplicates are the documents that are identical in content but they differ by the URL.A near duplicate document differs by a tiny fraction from the original one, it could be an image, text or even the order of the text. It is very important to have an algorithm that will detect both of these cases having a good time and space complexity.
We were given a dataset with fake and real news annotated from various fact checking sites[7] and we crawled and scraped the web to find near duplicate articles and label them correspondingly. For achieving this we used the simhash[8] algorithm.

From our tests we came to several conclusions regarding the impact of the search engine that has to the number of duplicate articles detected. Because the dataset comes from fact checking websites , the search engine firstly displays these websites, pushing any duplicate articles to a lower result position.Also there are many websites that reference fake articles , and as a result they are not considered duplicates by our hashing algorithm. Other issue is that a subject can be expressed by different words , and this is a weakness for locality sensitive hashing algorithms.

Additionally we have noticed a huge dependency on the hamming distance threshold. With a very strict threshold we have many true negatives , mainly because most of the times we deal with near duplicates, and also the parser may bring additional information, such as elements of the website.

2. Implementation

First, we used the simhash algorithm to create a fingerprint of the fake and true dataset that are given and store them in a database. Then, we crawled the web searching for similar articles based on the title of the article that we are going to compare. We used custom search engine API[9] provided by Google. The links returned by the API were downloaded and parsed using newspaper library[10] in Python. Again using the simhash algorithm, we generated a fingerprint for each of the article downloaded and we compared them using the simhash and considering a duplicate document distances lower than 5 bits.

2.1 SimHash

First, we create k-grams of the document. Through experimentation, we found that the optimal k is 5. Then, we use a cryptographic hash function to all the shingles. The purpose of the cryptographic hash function is to avoid collisions. In our case, we have used MD5. For each bit position we count the number of input hashes with that bit set (1) and subtract the number of input hashes with that bit not set (0). Once done, every position with negative counter is set to 0 in the result and for every other position is set to 1.

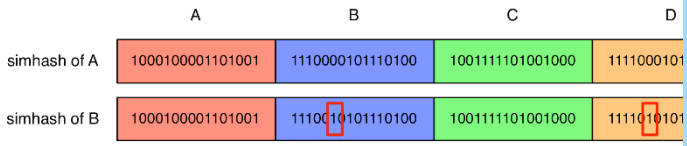
inputs	1	0	1	1	1	0	1	1
	0	0	1	0	1	1	1	0
	0	1	1	0	0	0	1	1
	0	1	0	0	0	0	1	0
	1	1	1	1	0	0	1	1
	1	0	0	1	1	1	0	0
	0	0	0	0	1	0	1	1
counters	-1	-1	1	-1	1	-3	6	1
result	0	0	1	0	1	0	1	1

To measure the similarity of two hashes we count the number of bits that differ between two queries as measure of dissimilarity. By doing this, we achieve dimensionality reduction because instead of storing hashes for each document, we aggregate them into one single integer that will represent all hashes as a fingerprint.

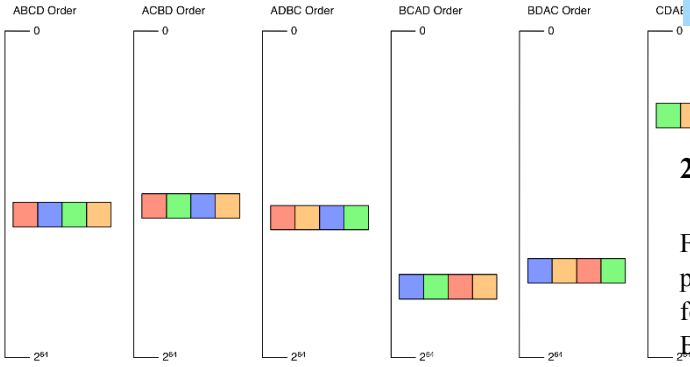
simhash of A	0	0	1	0	1	0	1	1
simhash of B	0	0	1	1	1	0	1	1
A ^ B	0	0	0	1	0	0	0	0

Bit population(A ^ B) = 1

In order to find similar hashes, we divide them into groups, as shown below. We notice that, for n different bits we have at most n different groups.



For each permutation we sort them and measure the bit difference between the neighbouring hashes with the same prefix in ascending order. This will cause the near duplicates to be close for comparison and the difference will be in the least significant bits.



The time complexity for this algorithm is O(d*ln(n)) where d is the number of permutations and ln(n) is the time complexity for comparing hashes that share the same prefix.

2.2 Data storage procedure in Database

For our purposes we have used MySQL database because it provides fast information retrieval and quality of service. We store the URLs searched in a table called urls with their simhash value and any duplicate urls matched from the web in a separate table called duplicate_urls. We try several number of bits as a threshold distance for each url we scrape and we keep them so we can extract information to compare how different values affect TN(True negatives) and FP (False positives). Due to limitations on the sql queries regarding the comparison of the hashes described above we have to load all the records to the main memory and then compare them with each article that google custom search has provided.

2.3 Web Crawling and web scraping

For data processing we used the newspaper[9] API for python. We took urls from our datasets and use the library for extracting features like the title and the text of the article. By using the library, we removed all the unnecessary elements that the web uses like tags, headers, photos and also some advertisements. There were also some problems with some of the links that couldn't be parsed by the library. We used the urllib[10] to compare the host name of the links that google custom search engine returned so that we won't mark the same document as duplicate.

3. Related work

3.1 Duplicate-detection systems

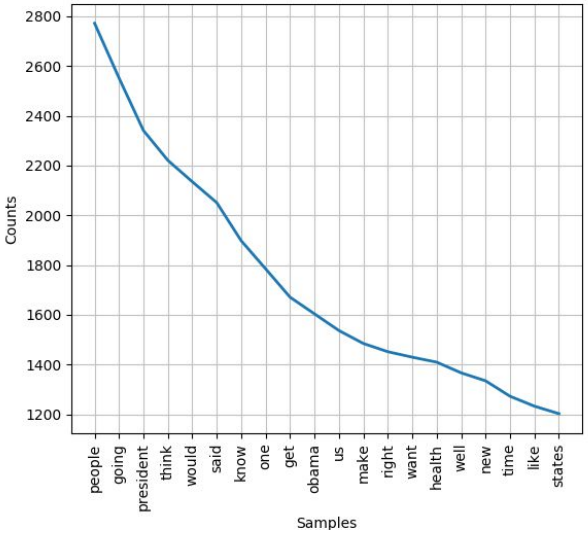
Duplicate detection systems are used for several types of documents collections.

One application of these systems is in **Web Documents** where we find related-pages[11], for identifying web mirrors. Our work is focused in this category for identifying near duplicate websites that are similar with our dataset.We see a use of these systems also for detecting near-duplicates for **spam detections**[12] in emails. Given a large number of recently received e-mails, the goal is to identify SPAM

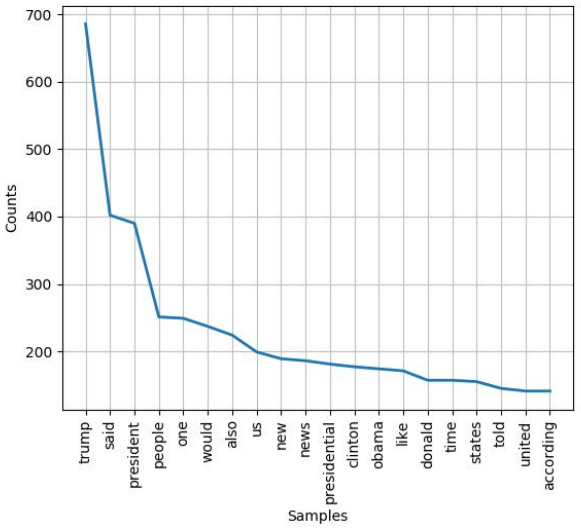
before depositing the e-mail into recipients mailboxes.Various algorithms are used also for near-duplicate detection **to reduce storage**[13] for files and indexes.Duplicates and Near Duplicate Web pages are creating large problems for web search engines like increasing the space needed to store the index, either slowing down or increase the cost of saving results and annoy the users. Elimination of near-duplicates saves network bandwidth, reduces storage costs and improves the quality of search indexes. It also reduces the load on the remote host that is serving such web pages.

4. Graph about most popular words

Frequency of words in True Dataset

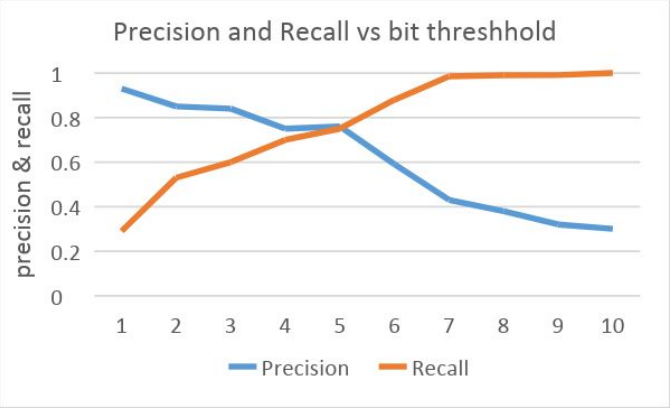


Frequency of words in False Dataset



Research has found that false political information tends to spread “3 times” faster than other false news[14]. As we see in the graph above the most frequent words have to do with the politics and especially with the most influential people because the fake news has a tendency to deceive people opinions. According to the dataset that is given we notice that the articles have to do with the presidential elections in America.

5. Evaluation



We have selected a sample from our dataset of 20 documents from which the half of them if searched on the web,some results will be duplicates. We tried with different bit to see how precision and recall varies to find the ideal number of bit and we came to the conclusion that the optimal bit value is 5. Precision and recall are defined by the following formulas.

$$\text{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \text{ or } \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

The bit threshold is affected by the fact that websites carry extra information apart from the main content which are also scraped causing the ideal bit threshold to increase. There were also some false positives from the websites that redirected the request to a specific page usually for authentication or subscription purposes.

Acknowledgements

We thank our professor George Pallis for teaching us important principles on locality sensitive hashing.We also thank our lab instructor Pavlos Antoniou for providing us the knowledge on data analytics tools. We thank also Demetris Paschalides for coordinating us for the whole project.

Code

Source code for SimHash at:
http://code.google.com/p/simhash/.

Libraries newspaper,urllib,nltk,mysql.

References

[1] Surname A, Surname B and Surname C 2015 *Journal Name* **37** 074203

[2] Surname A and Surname B 2009 *Journal Name* **23** 544

[3] Hunt, Elle (December 17, 2016). "What is fake news? How to spot it and what you can do to stop it". The Guardian. Retrieved January 15, 2017.

[4] Schlesinger, Robert (April 14, 2017). "Fake News in Reality". U.S. News & World Report.

[5] "The Real Story of 'Fake News': The term seems to have emerged around the end of the 19th century". Merriam-Webster. Retrieved October 13, 2017.

[6] Carlos Merlo (2017), "Millonario negocio FAKE NEWS", Univision Noticias

[7] www.politifact.com/, www.factcheck.org/, www.snopes.com/

[8] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.473.7179&rep=rep1&type=pdf&fbclid=IwAR3Q3lqbzQygOYtW6eqfSwlLi_DPOKXkzfbccWH9Tgw0rpRCQbGs01ma13Q

[9] https://pypi.org/project/newspaper3k/

[10] https://docs.python.org/3/library/urllib.html

[11] J. Dean and M. Henzinger. Finding related pages in the World Wide Web. Computer Networks, 31(11–16):1467–1479, 1999

[12] A. Kolcz, A. Chowdhury, and J. Alspector. Improved robustness of signature-based near-replica detection via lexicon randomization. In SIGKDD 2004, pages 605–610, Aug. 2004

[13] U. Manber. Finding similar files in a large file system. In Proc. 1994 USENIX Conference, pages 1–10, Jan. 1994.

[14] Vosoughi, Soroush. "THE SPREAD OF TRUE AND FALSE NEWS ONLINE" (PDF). MIT Digital. Retrieved 5 March 2019.