A PRACTICAL ALGORITHM FOR EFFICIENTLY DEDUPLICATING HIGHLY SIMILAR NEWS IN LARGE NEWS CORPORA

Wu Zhang

Miotech, 69 Jervois St, Sheung Wan, Hong Kong

ABSTRACT

Duplicated training data usually downgrades machine learning models' performance. This pa- per presents a practical algorithm for efficiently deduplicating highly similar news articles in large datasets. Our algorithm comprises three components - document embedding, similarity computation, and cluster- ing - each utilizing specific algorithms and tools to optimize both speed and performance. We utilize the Doc2Vec model to generate document embeddings.We employ Faiss for rapid similarity search. To perform clustering, we make use of the disjoint set data structure. We demonstrate the efficacy of our approach byaccurately deduplicating over 7 million news articles in less than 4 hours.

KEYWORDS

Document Embedding, Text Similarity, News Deduplication, Natural language processing

1. INTRODUCTION

Data duplication can negatively impact machine learning models by introducing data bias during model training and data leakage during model evaluation. While techniques existto identify and remove identical data, our algorithm aims to deduplicate highly similar texts. This is particularly relevant for news articles, which are frequently forwarded among sources with minor changes to the original text. Failing to remove these duplicates risks counting the same news multiple times, leading to models that do not generalize well.

Alexandra et al. [1] investigated the impact of text duplication on models and found that small amounts of duplication were tolerable, but substantial duplication was harmful. Lee et al. [2] demonstrated that removing duplicates from training data improved the efficiency of language models without compromising accuracy.

Unsupervised news deduplication poses two significant challenges. Firstly, the amount of news available is vast, making character comparison-based algorithms such as comput- ing the editing distance [3] computationally expensive for large news corpora. Secondly, duplicated news articles are not identical but differ only slightly, rendering hashing-based algorithms [4] unsuitable for this problem.

David C. Wyld et al. (Eds): SOEN, SIPP, PDTCA, ITE, CCSIT, NLPCL, DaKM, BIGML, AISC -2023
pp. 163-168, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131214

To illustrate the challenges of unsupervised news deduplication, consider the Chinese news example in Fig 1. The main content of both news articles is the same, but one contains additional source and copyright information. Such minor differences in duplicated news articles make deduplication challenging, emphasizing the need for our proposed algorithm. In this paper, we propose a practical and easy-to-implement algorithm to identify highly similar news articles in large news corpora. Our proposed algorithm accurately deduplicates over 7 million news articles in less than 4 hours, enabling downstream tasks reliant on news corpora to be more accurate and efficient.

2. OUR ALGORITHM

Our algorithm consists of three key steps: document embedding, document similarity computation, and clustering. In the document embedding step, each news article is converted

Fig. 1. A news duplication example where two news articles have the same title and main content.

into a fixed-length dense vector, enabling efficient computation of document similarity. The document similarity computation step scores each pair of news articles in the cor- pus, producing a similarity score. Finally, the clustering step uses the similarity scores to group highly similar news articles into clusters. The choices of document embedding and similarity computation are flexible so they are not specified in the code.

```
// Any document embedding function which produces fixed-length vector
function ComputeDocEmbedding(d)
    ...
    return Document embedding of d
// Any similarity function computing similarity score between two vector
function SimilarityScore(e1, e2)
    ...
    return Similarity score between e1 and e2
// Main function, ds is a list of news
function DeduplicateNews(ds, threshold)
    es ← empty list
    // Embedding computation steps
    forall d in ds do
        es.append(ComputeDocEmbedding(d))
    end
    dis_set ← DisjointSet(len(ds))
    // Computing similarity and clustering step
    forall i in range(len(ds)) do
        forall j in range(i + 1,len(ds)) do
            if SimilarityScore(es[i], es[j]) > threshold then
                dis_set.union(i, j)
            end
        end
    end
    dedu_corpus ← empty set
    // For each set, pick the parent element as representative
    forall i in range(len(ds)) do
        dedu_corpus.add(ds[dis_set.find(i)])
    end
    return dedu_corpus
```

Algorithm 1: News Deduplication Algorithm.

3. Implementation Details

In this section, we outline our choices for each step of the algorithm. For the embedding step, we use the Doc2Vec [5] model, although pre-trained models such as BERT [6] are also viable options. We opt for Doc2Vec because it requires significantly fewer computation resources and training time to develop an in-domain model, and its performance is satisfactory for our news deduplication task. Additionally, Doc2Vec handles lengthy documents with ease and performs online news deduplication tasks quickly and efficiently, without the need for GPUs. One downside of using Doc2Vec is that its embedding outcomes are not stable if different initialization document embedding vectors are used. However, we found that this issue does not significantly affect the result of similarity computation.

Cosine similarity is our chosen similarity function. We classify two news articles as highly similar if they receive a similarity score greater than 0.8, and group them into the same cluster. To rapidly calculate all similarity scores and quickly identify highly similar news articles, we employ faiss [7]. Faiss is exceptionally fast at identifying the top-*k* most similar pairs and can be accelerated using GPUs. After running faiss, we need only iterate over the top-*k* most similar news articles for each news article during clustering.

In the final clustering step, we use the disjoint set[1] data structure to group similar news articles, with the parent element serving as the cluster representative. Other heuristics, such as article length or publishing date, may also be used to select representatives. The size of a cluster, measured by the number of elements it contains, can serve as an indicatorof the news article's importance.

4. EXPERIMENTS AND RESULTS

We applied our algorithm to a large news corpus containing 7,193,990 Chinese news articles crawled from the web. We preprocessed the documents by segmenting them into words using jieba[2], a Chinese text segmentation tool. To train the Doc2Vec model, we set the vocabulary size to 1,500,000, the window size to 8, and the output vector size to 256. The probability mass graph of news length in the corpus is shown in Figure 2. The computation time for each step of our algorithm is listed in Table 1. The most time-consuming stepis training the Doc2Vec model on the entire corpus, which takes 11,989 seconds. Using GPU, the faiss algorithm takes only 1,968 seconds to compute the similarity scores, and weretrieve the top-64 most similar vectors for each news's embedding vector. The clustering step is very fast, taking only 44 seconds. In total, our algorithm accurately deduplicates more than 7 million news articles in less than 4 hours.

Table 1. Time Cost of News Depulication Algorithm.

Step	Time Cost (in Seconds)
Doc2Vec	11989
faiss (gpu)	1968
clustering	44
Total	14001

After applying our news deduplication algorithm, we were left with a total of 4,969,222 deduplicated news. The distribution of cluster sizes is provided in Table 2, and we found that the majority of clusters only contain one news article. To verify the accuracy of our algorithm, we randomly selected 5 clusters for each cluster size ranging from 2 to 10 and1 cluster for each cluster size ranging from 11 to 30. Only 3 clustering errors were found during this process, with incorrect clusters having sizes of 2, 18, and 23, respectively. From our analysis, we found that our algorithm is more likely to produce errors when the cluster size exceeds 30. To improve the precision, we can choose to discard all clusters whose sizes exceed a certain threshold.

[1] https://www.geeksforgeeks.org/disjoint-set-data-structures/

[2] https://github.com/fxsjy/jieba

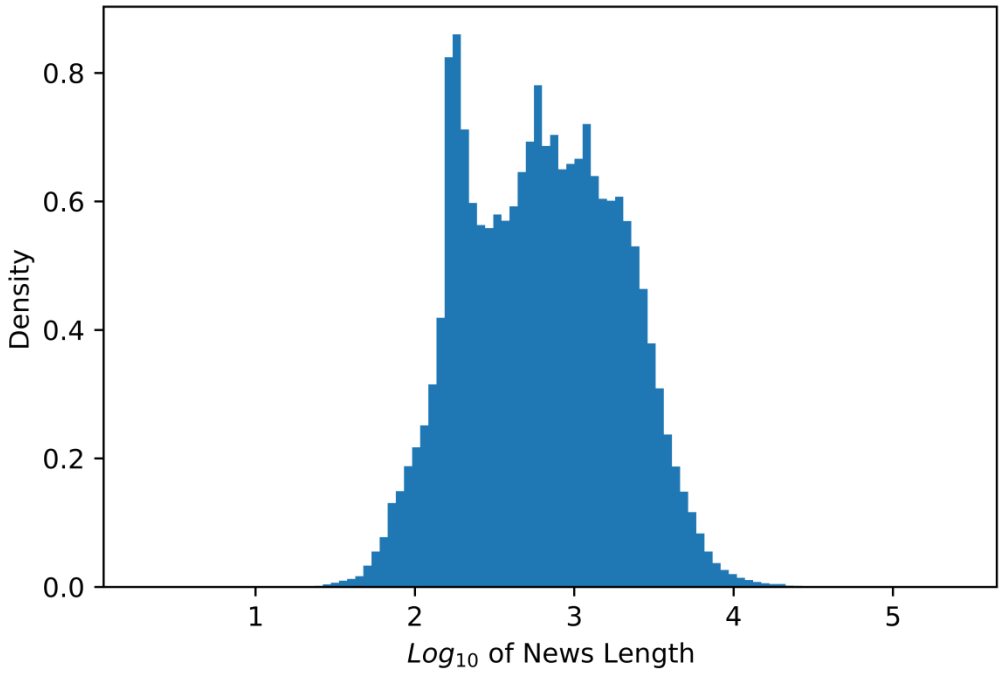

Fig. 2. Approximated News Length Probability Mass.

Table 2. Cluster Size distribution.

Cluster Size	1	2	3	4	5	6	7	8	9	10	> 10
Number of Clusters	3999816	655829	168218	70159	31266	16341	8842	5282	3332	2215	7922

5. DISCUSSION

To further improve the performance of our algorithm, we can explore using more stable embedding algorithms like Sentence-Bert [8] and . While Doc2Vec performed well for our news deduplication task, its embedding results are not always stable if using different initialization document embedding vectors.

In addition, to improve speed without sacrificing accuracy, we can utilize faiss's approximate similarity search feature. This can help us quickly find highly similar news, as these news are likely to have high similarity scores.

It is worth noting that highly similar news can provide a natural source of data dis- turbance [9] and improve the robustness of machine learning models. However, knowing which news articles are duplicates and how to group highly similar news can reduce the effort required for manual annotation and provide us with more options for using the data. There are also other heuristics that can be incorporated into the algorithm to improve its precision. For example, named entity recognition (NER) can be used to identify important entities in news articles and compare them to identify similarities. Additionally, clustering based on the publication date can be used to group news articles that cover the same event.

Lastly, one interesting challenge in news deduplication is dealing with template-style daily

news, where the articles follow a similar format and only differ in certain details,such as the names of people or locations. In such cases, the similarities between news are usually very high, making it challenging to develop a robust algorithm to handle them. Addressing this challenge and developing an algorithm that can accurately detect and group such news would be highly valuable for automating news deduplication tasks in a more comprehensive way.

ACKNOWLEDGMENTS

We would like to thank our web crawling team for crawling all the data that we need and thank our CTO Tao Liu for allowing us to disclose our algorithm.

REFERENCES

1. Alexandra Schofield, Laure Thompson, and David Mimno. Quantifying the effects of text duplication on semantic models. In Proceedings of the 2017 conference on empirical methods in natural language processing, pages 2737–2747, 2017.
2. Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. arXiv preprint arXiv:2107.06499, 2021.
3. Gonzalo Navarro. A guided tour to approximate string matching. ACM computing surveys (CSUR), 33(1):31–88, 2001.
4. Manreet Kaur and Jaspreet Singh. Data de-duplication approach based on hashing techniques for reducing time consumption over a cloud network. Int J Comput Appl, 142(5):4–10, 2016.
5. Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In International conference on machine learning, pages 1188–1196. PMLR, 2014.
6. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.
7. Jeff Johnson, Matthijs Douze, and Herv´e J´egou. Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3):535–547, 2019.
8. Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks.arXiv preprint arXiv:1908.10084, 2019.
9. Vukosi Marivate and Tshephisho Sefara. Improving short text classification through global augmentation methods. In Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, August 25–28, 2020, Proceedings 4, pages 385–399. Springer, 2020.

AUTHORS

Wu Zhang received Ph.d. from the University of Hong Kong. He did Bachelor of Science in Mathematics from the University of Science and Technology of China. Currently, he is working as a senior software engineer in NLP at Miotech. His research interests lie in the field of natural language processing, specifically in machine translation and information extraction.