Near Duplication Data Set Tool - Mini-project Report

Adi Simhi and Efrat Levkovizh

Abstract

The Web is the leading source for modern large-scale datasets for language models. Near duplication is a widespread phenomenon on the Web and, therefore, an issue in NLP datasets. In many cases, the contents of one web page are identical to those of another except for a few characters - say, a notation showing the date and time at which the page was last modified. This project addresses the classification problem of near-duplicate sentences by generating new datasets and providing codebase infrastructure for dataset generation. We offer classical solutions to near-duplication detection as part of our codebase and analysis of these solutions in our report.

1 The Near-duplicate Dataset

This project aims to create a classification data set for near-duplicate pairs of sentences. We used a text classification dataset as a base to create the dataset. The base dataset is a dataset of sentences, categories by subjects. Our output dataset includes positive labeled samples, which are pairs of sentences, original and transformed, originated from the base dataset. And negative samples, which are pairs of sentences from the base dataset labeled as the same category.

1.1 Positive samples

The positive samples are pairs of sentences originating from the same sentence. One is left the original one, and the second is modified. The modifications we used are:

- Switching Synonym - Switches random words for one of their synonyms. We used WordNet and the NLTK (Wagner, 2010) library for the implementation of that part .
- Punctuation Insertion - adds random punctuation and numbers in random locations using mapping of punctuation to distributions. The mapping is configurable.
- inject Typos - Inject typos such as deletion, replace of letters and insertions of letters in random places. Replacement, deletion and insertions probabilities are configurable by letter.

1.2 Negative samples

We create the negative samples by taking pairs of sentences with the same subject category in the base dataset. The motive is that these sentences would probably have similar embeddings, but they are not near duplicate. This way, we hope to achieve a higher difficulty level. Negative samples can also be pairs of sentences not from the same category in the base dataset; their distribution is configurable.

2 Place within BigScience

This problem is related to the group of data-tooling in the BigScience community. We contacted them and informed them that we were working on that project. This project idea is from the bigsceince-project-students slack channel, suggested by Huu Nguyen and Yonatan Belinkov. Upon finishing documentation and configuration development, we will send the group a link to our open GitHub repository. We hope it will serve them well as a near duplication dataset generator or/and as an infrastructure for future toolings.

3 Background

The problem of near duplication is very much alive on the Web, which contains multiple copies of the same content. By some estimates, as many as 40% (Schütze et al., 2008) of the pages on the Web are duplicates of other pages. Search engines avoid indexing multiple copies of the same content to keep down storage and processing overheads. Web content commonly functions as a source for NLP datasets for massive pre-trained language models. These duplicates and near-duplicates are slipping into the model training process and possibly cause biases, and some are unwanted. Therefore, deduplicating the datasets is a significant preprocessing stage.

Over the years, there have been numerous works presenting models that are trying to detect near duplication, such as (Wu et al., 2011), and (Rodier and Carter, 2020). We want to help these efforts by providing a near duplicate data set tool.

4 Contributions

Our contribution is a tool for generating a near-duplicate dataset using a base classification dataset. It is easy to use and has a lot of configurable freedom. We tested it with several classification methods. You can find examples in the table below. We tested our datasets with several solutions and evaluated their performance to get a sense of the difficulty of our new datasets.

5 Results

To assess the dataset, we used similarity score methods and distance metrics. We determine the threshold using a train set. We ran a classification fold (number splits=5). The classification data set we used is taken from `https://www.kaggle.com/amananandrai/ag-news-classification-dataset`.

5.1 Solutions

Deduplication of a data set is a heavy computational process. Therefore many algorithms are not feasible for actual use. We tested similarity algorithms without taking into account their feasibility in the actual run.

Models of similarity tested are:

- Cosine similarity score of sentence-BERT (Reimers and Gurevych, 2019) embedding of the two sentences.
- Jaccard similarity between the two sentences.
- Edit similarity - edit distance is the percentage of the minimum number of operations required to transform one string into the other out of the length of the sentences, which means that the edit similarity is one minus the edit distance.
- Overlap similarity - the percentage of words that are similar.

5.2 Analysis

Model	F-score	Accuracy
Sentance-Bert	**0.575**	**0.904**
Jaccard similarity	0.032	0.516
Edit similarity	0.537	0.867
Overlap similarity	0.25	0.644

The above table displays the test result of the near duplicate classification problem using f-score and accuracy metrics. As can be indicated by the results, the best method to detect duplication is sentence-BERT (Reimers and Gurevych, 2019). Moreover, we see that no model performs excellently on the data set.

6 Code

The code is accessible by this GitHub link `https://github.com/EfratLe/NLP-seminar-mini-project`. The repository contains a Readme file that describes the command line format, and the configuration file options.

7 Conclusion

In this work, we created a dataset to the known problem of finding near duplication in text. The generated dataset is highly dependent on the base dataset, and we concluded that our generated dataset is challenging by testing it.

Future work ideas:

- Further testing using used tools for deduplication can better picture their accuracy and the dataset quality.
- Making near duplication datasets of real-world data. Maybe by scrapping text through the web. Some web data tends to contain near duplication- dates and other dynamic data.
- Developing an accurate and feasible method for detecting near duplication.
- Creating different levels of complicated data sets with a variety of labels.

We hope that this work will help test and develop better models to test near duplication in text.

References

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. pages 3973–3983.

Sentence 1	Sentence 2	label (1 - near duplicate)
Bryant Jury Selection Behind Closed Doors (AP)	Bryant Jury Selection Behind close door (AP)	1
Anti-Spyware Spending Set To Skyrocket	Anth-Sphyware spending nsetw To Skyrocket	1
"Mbeki calls for arms embargo, rebels.."	Mbeki calls for wewaponry mbargo, rebEls.."	1
Miller slams transit funding	Passenger Revenue Boost for Air France-KLM	0

Figure 1: Examples from the new dataset.

Simon Rodier and Dave Carter. 2020. Online near-duplicate detection of news articles. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 1242–1249. European Language Resources Association.

Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge.

Wiebke Wagner. 2010. Steven bird, ewan klein and edward loper: Natural language processing with python, analyzing text with the natural language toolkit - o'reilly media, beijing, 2009, ISBN 978-0-596-51649-9. *Lang. Resour. Evaluation*, 44(4):421–424.

Yan Wu, Qi Zhang, and Xuanjing Huang. 2011. Efficient near-duplicate detection for q&a forum. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 1001–1009. The Association for Computer Linguistics.