Research and implementation of methods for plagiarism detection

Inessa Alaverdyan

September 2020

1 Introduction

Plagiarism is defined as using an idea derived from an existing source as new and original. There are different types of plagiarism used in academia and not only, and this report is concerned with plagiarism detection on fully identical, almost identical, synonymous, paraphrased sentences. Other than the above mentioned ones it is also concerned with detecting plagiarized sentences with word insertion, deletion and substitution to the original sentence.
Fully identical plagiarism is a word-for-word transcription of the original text, generally it is the easiest one to detect, as a simple character-by-character comparison will produce favourable results. Another very similar type of plagiarism is the almost identical one, in which the copied sentence is edited by insertion, deletion or substitution of a few words or punctuation marks. Another widely used plagiarism technique is to change words with synonyms or negated antonyms, or to paraphrase the entire sentence by using a different sentence structure and vocabulary.
The goal is to implement a method to detect whether the given two input sentences are plagiarized or not. If the input sentences convey similar meaning the resulting output of our method should be True, otherwise it should be False.

2 Description of the Solution

2.1 Jaccard Similarity

Jaccard score measures the similarity of two sentences. It is defined as the size of the intersection divided by the size of the union of two sentences.[1]
In the calculate_jaccard_score() method, we have two sentences as our inputs. We first lemmatize all the words in those sentences for comparison. Then we calculate the jaccard_score by definition. In the final step, we just put a threshhold to determine whether the results generated by this method are true or false. In my code I used 0.4 as my threshold, which was chosen after some adjustments. An attempt to implement jaccard_score from sklearn libary was erroneous, the

results were far from correct. A more detailed explanation with code is presented in this colab.notebook.

2.2 Fuzzy-Wuzzy Similarity

This is a great improvement to the previous method. In the token_set_ratio method we first tokenize both strings, then split them into two groups: intersection and remainder. We use those sets to build up a comparison string. Then we compare strings of the following form:

s1 = Sorted_tokens_in_intersection
s2 = Sorted_tokens_in_intersection + sorted_rest_of_str1_tokens
s3 = Sorted_tokens_in_intersection + sorted_rest_of_str2_tokens

After which each pair is compared using fuzz.ratio, which computes the standard Levenshtein distance similarity ratio between two sequences, and the maximum value is selected as our similarity rate. [2] [3] As in the previous case, we need to select a threshold for determining sentence similarity, in my experiments *80* was selected as the empirical threshold.

2.3 Similarity using WordNet

Inspiration for this approach was taken from an article [4]
The idea of this algorithm is as follow: the sentences are first split into separate words and the words are wrapped into synsets from the WordNet library. Separate synsets can be compared. Similar synsets result in a higher score. We need to devise a method for comparing the whole sentences. In this algorithm every word from the first sentence is compared with every word from the second sentence. The words with maximum similarity are chosen and average is calculate.
The threshold was chosen experimentally to give results as good as possible.

There are many ways of potentially improving the algorithm. We can choose a different similarity measure (instead of the maximum and the average). We can also extract nouns and verbs and put only those words to the WordNet library. (The later approach was attempted but it didn't give better results on the analyzed dataset.)

3 Experiments and Analysis of Results

For analysing the algorithms implemented in this work Microsoft Research Paraphrase Corpus (MSRP) dataset was used.
In all three cases the algorithms predicted with a precision of less than 72% which is not a very favourable result.

In the table below in we can see the predicted results by each of our algorithms and in each consecutive column the truthfulness of those results.

	Quality	#1 String	#2 String	result_jaccard	Right_Results_jaccard	result_fuzzy	Right_Results_fuzzy	result_wordnet	Right_Results_wordnet
0	1	Amrozi accused his brother, whom he called "th...	Referring to him as only "the witness", Amrozi...	True	True	True	True	True	True
1	0	Yucaipa owned Dominick's before selling the ch...	Yucaipa bought Dominick's in 1995 for $693 mil...	True	False	False	True	False	True
2	1	They had published an advertisement on the Int...	On June 10, the ship's owners had published an...	True	True	True	True	True	True
3	0	Around 0335 GMT, Tab shares were up 19 cents, ...	Tab shares jumped 20 cents, or 4.6%, to set a ...	True	False	True	False	True	False
4	1	The stock rose $2.11, or about 11 percent, to ...	PG&E Corp. shares jumped $1.63 or 8 percent to...	True	True	False	False	True	True

Other than just experimenting the outcomes of the algorithms, we wanted to see if what was implemented was anyhow better than just saying that every sentence in the dataset is a paraphrase.
Apparently the three presented algorithms did not perform much better than just stating that every 2 sentences are a plagiarism. To assess their value one could try to run them on a different dataset with possible simpler paraphrases. The MSRP dataset contains very long and difficult sentences even for a human. Unfortunately, none of the presented algorithms understands the meaning of the presented sentences, so it cannot handle a true paraphrase where completely different words are used to describe the same idea.

4 Conclusion

In this report, we went through three simple methods of plagiarism detection. Unfortunately, none of them performed well. It can be attributed to the fact that the analysed dataset contained difficult sentences. Even much more advanced methods, as the ones mentioned in this paper [5], don't perform much better.