Plagiarism Detection Report

Sunday, 25.04.2021

P Rahul Bharadwaj

Problem Statement

The objective of this assignment is to develop an algorithm to detect the similarity index between two documents. The algorithm should take two different documents as inputs and process them to quantify the amount of text similarity in the documents.

Software Requirements

- **Nltk**
- **Python**
- **Numpy**

Algorithm

- The features of sentences can be roughly determined by nouns. So, we first decompose each text belonging to a group of target texts (texts of inspection target) and reference texts (texts which may be plagiarized), and generate noun only arrays from them in this process.

- We name the noun-arrayed target texts set as (Nt) and the noun-arrayed reference texts set as (Nr), and the elements of each set are as (nt, nr). From here we simply call nt the target, and nr the reference.

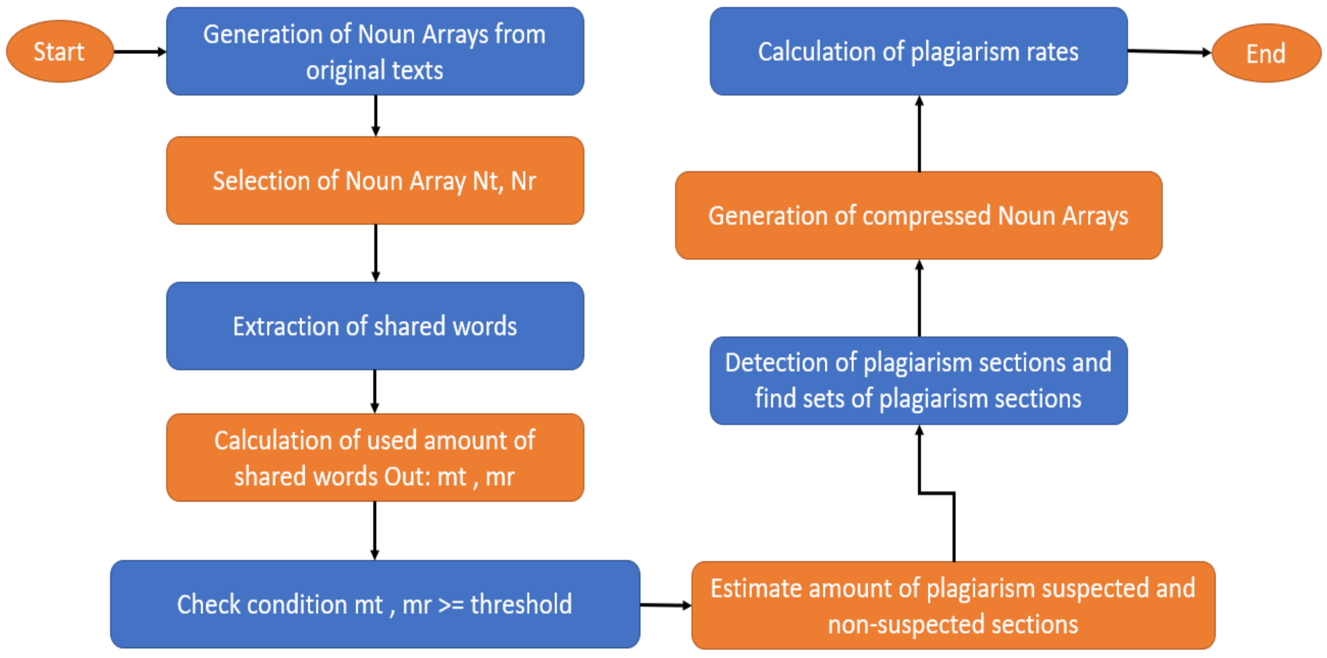
- In this process, we extract shared words between noun arrays nt and nr, generated in the previous process, and name the set of them as (C).

- We calculate the used amount of shared words in the set C within noun arrays nt and nr using equation (1), and name them as mt and mr respectively.

$$mt = \sum_{k=1}^{N} f(nt_k, C)$$
$$f(nt_k, C) = \begin{cases} \text{String Length of } nt_k & (nt_k \in C) \\ 0 & (nt_k \notin C) \end{cases} \qquad (1)$$

Similarly mr is calculated by changing ntk and N for nr in equation (1) as well. After calculations of mt and mr, we have to introduce a numeric parameter which corresponds to "predetermined length" in our definition of plagiarism, as a threshold to determine whether plagiarism or not.

This is a flow diagram representation of our algorithm.


Start
Generation of Noun Arrays from original texts
Selection of Noun Array Nt, Nr
Extraction of shared words
Calculation of used amount of shared words Out: mt , mr
Check condition mt , mr >= threshold
Estimate amount of plagiarism suspected and non-suspected sections
Detection of plagiarism sections and find sets of plagiarism sections
Generation of compressed Noun Arrays
Calculation of plagiarism rates
End

- **we then decide plagiarism suspected section and non-suspected section using nt, nr and C. The non-suspected sections on nt and nr are compressed in later processes.**

Below is the explanation of how we decide these sections.

Step 1:

We generate flag arrays Ft and Fr having the same length as nt and nr respectively, and name them as“plagiarism-suspected flag”. If the element of nt or nr is also an element of C, the corresponding index element is set “True”, otherwise “False” respectively. That is, the shared word locations on Ft and Fr are “True”. The indexes which were set "True" in this process become plagiarism suspected section for the time being.

Step 2 :

We convert the isolated “False” part to “True”. The isolated "False" means a point, at which the both front and back are "True" on the flag arrays Ft and Fr. At the same time, the elements at the same index on the corresponding arrays nt and nr are changed to wildcards (#) respectively. The aim of this operation is to counter the camouflage that makes the plagiarism section be shown shorter by dividing longer section using swap of word and so on

Step 3 :

We convert the small "True" sections to "False". A small "True" section means that elements of the plagiarism-suspected flag array is continuously "True", and also the total string length of elements in the noun array is less than M in the same "True" indexes. By setting these sections as "False", the compression ratio in the next process becomes high, and as a result, the amount of calculation can be reduced. Also, if all the elements of Ft or Fr are "False" in this process, it is obvious that there is no prospect to detect plagiarism section

- we then generate compressed noun arrays for comparison. The arrays nt and nr are compressed and converted according to Ft and Fr respectively. The elements of nt and nr are left if the corresponding elements of Ft and Fr are "True" respectively, because they are plagiarism-suspected sections. On the other hand, the elements of nt and nr are compressed into one element (space) if the corresponding elements of Ft and Fr are "False" respectively, because they are not plagiarism-suspected sections. We name these compressed noun arrays as nt* and nr* respectively.

- In this process we next apply the basic detection method (See figure below) to the compressed array nt* and nr*. We shift one of the nt* and nr* by one index at a time, and try to detect the matching section of words. For matching process, the wildcard (#) which introduced earlier is considered to match any word. If matching is detected, continuous matching word sections are treated as one set, and the string length of each set is calculated separately. And if the calculated value is M or more, the set (partial array) is determined to be "plagiarism section"

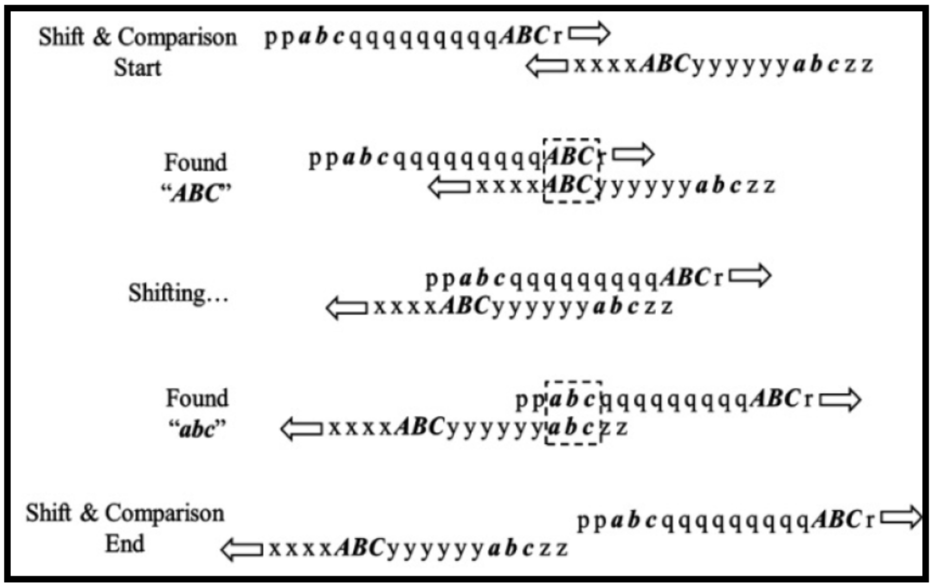


Fig. Process of Basic Detection Method of Plagiarism Section

Here, we name a set of registered "plagiarism sections" as Sp, and its elements as sp which are noun arrays. we calculate the plagiarism rate of nt using itself and Sp which is a plagiarism section set. Of course, All elements in Sp are partial arrays of nt. Assuming that the total string length of the array elements in nt, that did not match any array sp, is lo, and the total string length of all elements in nt is l, the plagiarism rate of nt is given by equation (2) .

$$Rp = \frac{l - lo}{l} \qquad (2)$$

Conclusion

The plagiarism rate can be used as an indicator when deciding whether to reject the report. For example, it is possible to eliminate reports that are not worth reading automatically by deciding that the plagiarism rate is 50% or more as a rejection.

References

- Plagiarism Detection Technique

  A Detection Method for Plagiarism Reports of Students ,Authored by Daisuke Sakamotoa, Kazuhiko Tsudab