The Porter stemmer algorithm is a broadly used, however, an essential tool for natural language processing in the area of information access. Stemming is used to remove words that add the final morphological and diacritical endings of words in English words to their root form to extract the word root, i.e. called stem/root in the primary text processing stage. In other words, it is a linguistic process that simply extracts the main part that may be close to the relative and related root. Text classification is a major task in extracting relevant information from a large volume of data. In this paper, we suggest ways to improve a version of the Porter algorithm with the aim of processing and overcome its limitations and to save time and memory by reducing the size of the words. The system uses the improved Porter derivation technique for word pruning. Whereas performs cognitive-inspired computing to discover morphologically related words from the corpus without any human intervention or language-specific knowledge. The improved Porter algorithm is compared to the original stemmer. The improved Porter algorithm has better performance and enables more accurate information retrieval (IR)

Keywords: stemming algorithm, natural language processing, information retrieval, APSA, Porter algorithm

UDC 004
DOI: 10.15587/1729-4061.2021.225362

DEVELOPMENT FOR PERFORMANCE OF PORTER STEMMER ALGORITHM

Manhal Elias Polus
Postgraduate Student*
E-mail: manhal.programmer@gmail.com
Thekra Abbas
PhD, Assistant Professor, Head of Department*
E-mail: thekra.abbas@uomustansiriyah.edu.iq
*Department of Computer Science
College of Science
Al-Mustansiriyah University
Palestine str., Baghdad, Iraq, 14022

Received date 08.12.2020
Accepted date 08.02.2021
Published date 26.02.2021

1. Introduction

Removing suffixes by automatic means is an operation that is especially useful in the field of information retrieval. With the huge increase in data in our digital age, it is difficult to recover data by traditional methods, and its great importance is considered as an essential tool for all search tasks on the web and requires to retrieve information. Text extraction algorithms enable the user to specify the required information from among a huge collection of data, and one of the most important factors that must be available information retrieval (IR) systems and natural language processing (NLP), must be fast and accurate. Preprocessing includes three common steps before progress to perform any operation on the data such as tokenization, stopword removal and stemming. These methods enable us to remove text data that contains some formats such as symbols, date, time and numbers, which amount to 80 % in one document. It uses a Word Stem mechanism to convert the word shape to its basic form, for example, accept←acceptable, acceptance, acceptances, acceptation, accepted, accepting. The basic text serves information retrieval systems to access relevant and modified documents at the frequency of the term leading to access to relevant documents at higher levels. In addition, reduces the stem by switching morphology variables to one root and helps reduce storage space and improve search efficiency.

This paper aims to get the origin of words that has meaning, the main purpose is to improve the performance of information retrieval by removing suffixes and prefixes from words and to conserve time and memory and hence reduce the size and complexity of the data in the system by reducing the size of words. The system uses the improved Porter stemmer technique for word pruning.

The essential word form in many languages is ameliorated to form various word forms according to the role of the word in a sentence. The word is created through various linguistic processes, for instance, combination of minimum two words, affixation, adding a prefix and/or suffix, and creation of a new vocabulary of existing words. Stemmer is the algorithm of reducing a derived word to the word stem, where different morphological forms of words are connected to their stem (essential word). It is usually satisfactory that related words are linked to the same stem (root) [1], even if this stem is not valid. Stemmer is the simple form of language that is most useful in languages with a complicated morphology, where a single word has a large number of variants. A huge variety of methods and algorithms have been suggested for this function. Moreover, the creation of fully Unsupervised Language Autonomous Stemmers is a major challenge.

2. Literature review and problem statement

The paper [1] identifies four different types of adhesive replacement algorithm for Porter, Lovins, S-Removal, and Paice, using a Hamming distance scale. The similarity and strength of each algorithm are pictured after making a list of 49,659 frequently used English words that originated from Mumby corpus and the UNIX spelling dictionary. The number of dimensions is six and the strongest stem is the one that has the largest value of each. According to their investigations, the main ones are Paice, Lovins, Porter, and S-Removal. The

results show that Paice had the strongest value in calling and compressing the pointer while its accuracy was the lowest.

The paper [2] presents a stem method applied to Persian/Persian literature, the stem was based on morphology and the stem suffix and the stem prefix were applied. The first step in this task was to have the last subroutine for a word that was already in the Farsi/Farsi adverb prefix list, then raise the suffix and in different cases of suffixes, the above component stem would define a suffix that would give a word in fewer letters. In other words, the suffix can be one or more characters, and when the other suffix is added, a longer suffix will be produced and will be canceled. The algorithm was developed to handle some issues and exceptions in the database such as words that were structurally similar to other different words. Likewise, the algorithm can find literal suffixes, but it will not cancel them. But there were unresolved issues like the "stan" suffix has not been removed because it is of common use among countries. In addition, the algorithm was limited, and the stem should have three or more characters after removing the suffixes and prefixes. If it is less than three characters long, the algorithm removes the suffix considering that the result will not be less than three and a portion of the suffix will remain with the stem. Moreover, they used the BNF machine to implement this algorithm. The prefix is removed on two occasions, it is found and canceled, while the suffix will be deleted in 15 cases. Unlike the suffix stem, the word before deleting any prefix or suffix in each step. The word type is checked for suffixes and prefixes that were omitted in the previous steps.

The paper [3] proposes a query-based derivation system that gives formal variables that are accurately identified with query words. This method greatly improves retrieval performance by reducing the impact of those specific variables that are not related to the initial query's intent. The derivation method greatly increases accuracy due to a small decrease in the recall. The method runs in two steps. Through the first step, words are classified into the group on the basis of the most common standardized prefix using the modified Maximum Information Interchange (MMI). The groups created in the first step are applied as training data for the maximum entropy classifier in the second step for selecting when and how to cut a word.

The paper [4] proposed a method for improving applications related to distributed computing that are characterized by their adaptability and flexibility with users. At the same time, information retrieval (IR) is often defined in terms of location and document delivery to users to meet their information requirements. Often the morphological variants of each word contain a similar semantic interpretation and are also considered equivalent for infrared applications. The CAS algorithm was introduced and proposed and is a modified version of the commonly used Porter derivative. As for the derivative words with meanings only, they are considered as outputs of the stem. The results of the modified algorithm showed that it significantly reduces the error rate in the Porter's algorithm and is approximately from 76.78 % to 6.76 %, which gives us a good insight in terms of effectiveness.

In [5], the researchers used morphological methods when they analyzed approximately 80,293 news articles published in the Turkish daily Milliyet. The training totals were estimated at 5,000 documents, each of which has approximately 500 words, and were manually checked, analyzed and distinguished. They analyzed the object using the K-Nearest Neighbors, Naive Bayes, Support Vector Machine algorithms. The results of the three different classifiers showed that each one has special effects on morphological information when texts in the Turkish language are classified. All the above-mentioned classifiers also showed that five-letter words with nearly six thousand features or more got the best scores, 92.5 %, 94.37 % and 93.12 %, respectively. But the operation of the classifiers and their effect on derivations are not mentioned.

The researchers considered stemming to be an important tool for almost two reasons. First, the stemming algorithm is used as a tool for improving retrieval precision by decreasing the length of the word by extracting the root from the word. Second, the stemming algorithm reduces the size of the indexed file, when the same root is extracted from different words. For example, connect, connection, connections, connecting, the extracted root is connect for all these words. Stemming rules will help to define which word types are linked to each other, and these associations may use the stemmer to determine the meaning of the word. By implementing six sequential steps, suffixes will be removed according to the rules of each step. The major drawback in the Porter stemmer algorithm is that the stem created after the six sub-steps is not always a real word. Sometimes, ambiguous words are produced after removing suffixes.

As can be seen from the above works, there are several major drawbacks of the Porter stemmer algorithm. First, the stem created after the six sub-steps is not always a real word. Sometimes, ambiguous words are produced after removing suffixes, when letters are omitted from the original root of the word resulting in a different word or a word without meaning. For example, in the word "general", the Porter stemmer removes "al" and it becomes "gener". Second, it does not deal with prefixes. The Porter algorithm cannot remove the prefix if it is present in the word. For example, prefixes: -ed such as decode, decrease. Third, the Porter stemming algorithm does not deal with irregular words, which causes a problem with the meaning and spelling of the words, for example, arise, arose, arisen.

3. The aim and objectives of the study

The aim of this study is to make the algorithm out of its restriction and drawbacks at the same time, by improving the algorithm, by adding 3 new rules to the existing six rules that increase the efficiency and accuracy of the algorithm. To achieve the aim, a database of new rules should be created.

To achieve the aim, the following objectives are accomplished:

– to create a list containing real word stems, such as arrive, secure, professional, ..., etc.;

– to create a list containing word prefixes;

– to create a list containing irregular words.

4. Materials and methods

Theses were collected from the year 2014 until 2019, these files were grouped in several classes. After the classification process, it is necessary to extract texts as the system deals with different formats of files such as IMG, PDF, DOC. The extracted texts are kept in a database, prepared for the stage of pre-processing. The pre-processing stage includes three steps. The first step is tokenization, the second step is stopword removal. The third step is to implement the improved stemming algorithm, which is explained in detail in section 6. The system was programmed by Matlab in the Windows 10 environment, specifications Core i7, RAM 12 GB, CPU 1.80 GHz.

5. Truncation algorithm

It is a way to remove the affixes (suffix or prefix) of words. Various types of stemming algorithm/truncating method are shown in Fig. 1 [6].

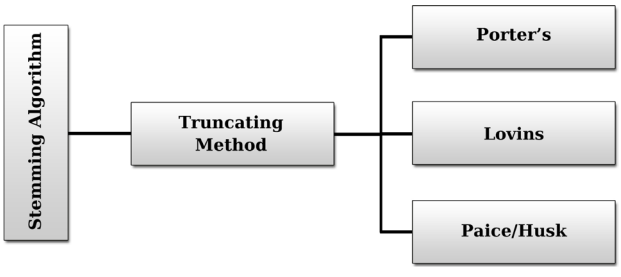

Fig. 1. Types of stemming/truncating method

It truncates with keeping the letters and removes the rest of the word depending on some rules and conditions. The truncation stemmer is divided into three types: Porter stemmer algorithm, Lovins algorithm, Paice/Husk algorithm.

5. 1. Porter stemmer

Porter stemmer is one of the most widely used truncation stemmers. Porter stemmer algorithm eliminates a word over a number of iterations before all rules or conditions are considered. Because it works without a lexicon and does not understand the context of a word, it is prone to some errors [8–10]. Words that have different meanings are reduced to the same stem. While words having similar meanings cannot be reduced to a specific stem at all. The study of the Porter stemmer shows that its performance was one of the best in terms of information retrieval recall/precision. Six sub-phases of the Porter stemmer algorithm are shown in Fig. 2 [10].

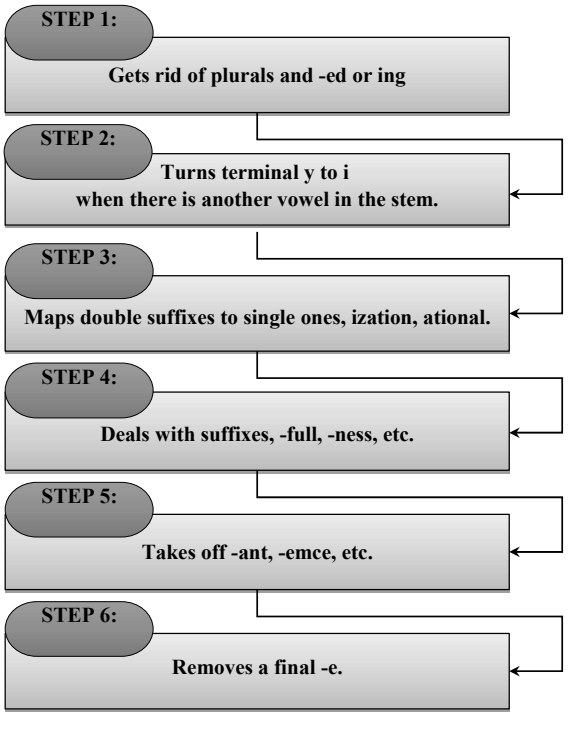

Fig. 2. Porter stemming

Step 1 is designed to deal with past participles and plurals. This is the most complex step and is divided into three parts in the original definition, 1a, 1b and 1c. Step 1 also removes inflectional suffixes (i-suffixes).

a) Step 1a:
SSES->SS
caresses->caress;
b) Step 1b
(*v*) ING<-
opening->agree;
c) Step 1c
(*v*) Y->I
history->histori
-Step 2.

This step is much more straightforward. It deals with pattern matching on some common suffixes. It removes derivational suffixes (d-suffixes) and follows some rules such as:

(m>0) ATIONAL->ATE relational->relate
-Step 3

deals with special word endings. It also removes derivational suffixes (d-suffixes). Composite d-suffixes are reduced to single d-suffixes one at a time. Therefore, if a word ends with -icational, Step 2 reduces it to -icate and Step 3 reduces it to -ic. Below is an example of rules applied in Step 3.

(m>0) NESS<-
possibleness->possible
-Step 4

checks the stripped word against more suffixes in case the word is compound. It deals with -ic, -able, -ive and many more, which are similar in strategy to step 3. An example of rules involved in this step is as follows:

(m>1) MENT<-
adjustment -> adjust
-Step 5

tides up the algorithm after removing suffixes in the previous steps. It checks if the stripped word ends in a vowel and is fixed appropriately. It consists of Step 5a and Step 5b, as indicated in the example:

a) Step 5a
(m>1) E
probate->probat;
b) Step 5b
(m>1 and *d and *L)->single letter
bill->bil.

6. The proposed approach

In order to minimize the noise present in any document and filter out unuseful words, it is necessary to pre-process the document. The proposed approach involved three processes as shown in Fig. 3, including tokenization, stopword removal, stemming algorithm.

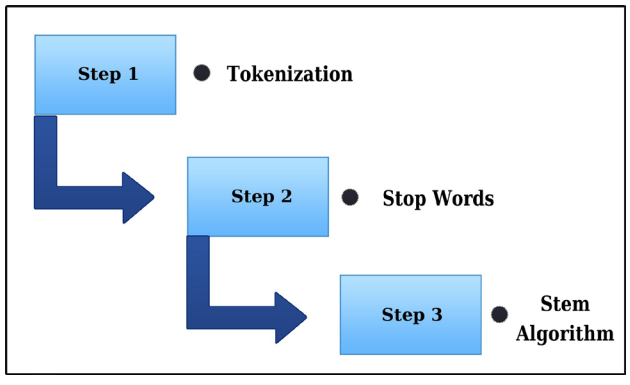

Fig. 3. Illustration of the proposed approach

6. 1. Tokenization

Tokenization is the first preprocessing, which is a critical activity in any information retrieval (IR) model. It simply separates all the words, numbers, and their characters, etc. from a given document, and these identified words, numbers, and other characters are called tokens. Fig. 4 shows the steps of tokenization.

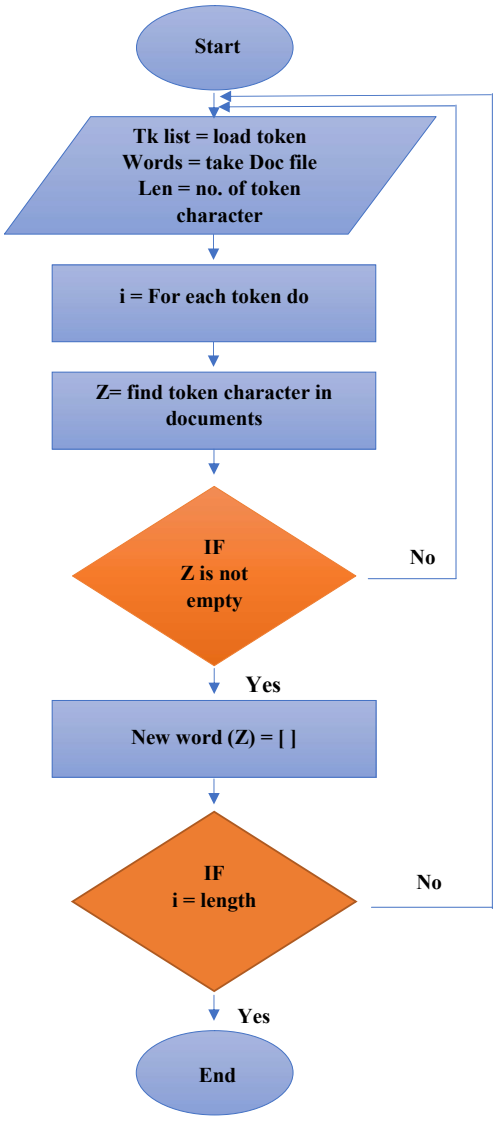

Fig. 4. Tokenization processing

The tokenization process includes six steps as follows:

– step 1: create a list that contains the number of tokenization, the file of words, the words '>' , '<' , '=' , '+' , '-' , '*' ,... etc. are saved in the specified file;

– step 2: load the first token in a variable;

– step 3: in this step, the first variable in the token list is compared with the document length, if the token matches the document length, move to the next token;

– step 4: if the document does not match the token, then take another token and check the document. In the case when a token is equal to the length of the token file, move to the next step;

– step 5: in this step, we get the new file that contains words without a specific token;

– step 6: check if the contain more of tokens, move to the first step and processing file, and in the event that processing is completed, it will exit the algorithm.

6. 2. Stopword removal

It is part of a natural language (NL), many words repeat in the document very frequently, but are actually unmeaning as they are applied to connect words together in a sentence. Removing stopwords decreases dimension spaces. The most common words in text documents are prepositions, articles, etc., which do not give the meanings of the documents. Fig. 5 shows the processing of stopwords.

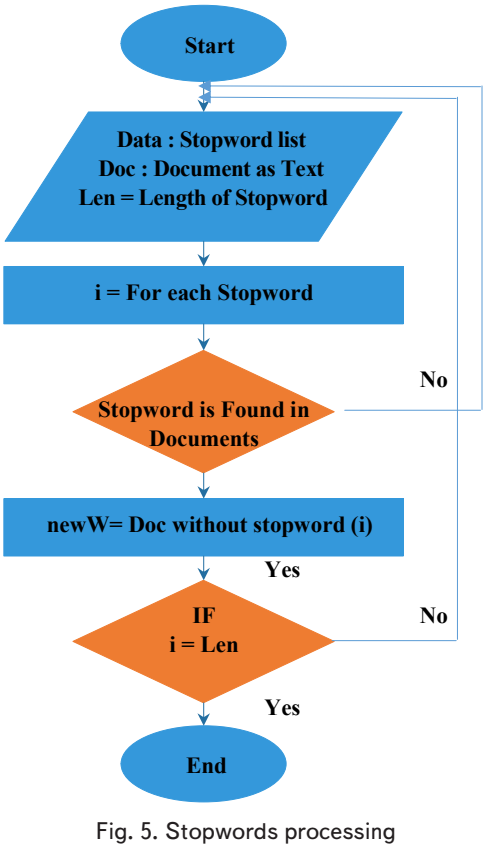

Fig. 5. Stopwords processing

The five steps included in the stopword removal function are as follows:

– step 1: create a list that contains a number of stopwords, including the words "a, in, or, the,... etc.", considering the reading and writing format, that is each word is one on one line, and after that the words are saved in the specified file;

– step 2: load the specified file;

– step 3: verify the text through a condition of the words in the document, if the words in the specified file match, move to step 4;

– step 4: save the text in a new document free of stopwords and then move to step 5;

– step 5: if the length of the specified file matches the new document, the process of removing stopwords ends and move to the next stage, which is stemming.

6. 3. Ameliorated stemming algorithm

The Porter stemming algorithm was developed (APSA) for the English language stem. After studying morphology, we used its properties to create the ameliorated stemmer and its importance in information retrieval (IR). It is more important in the field of essential words, as the algorithm became the standard for the English language. The Porter stemming algorithm has been improved by adding three sub-steps that include processing. Fig. 6 illustrates the steps of the ameliorated Porter stemming algorithm. Most of the anomalies were prepared in the English language for this purpose. Table 1 shows some examples of words.

Table 1

Examples of anomalous words that have been returned to their source

No.	Term	Derivative Term	Source
1	Abide	Abode/Abided	Abode/Abided/Abidden
2	Alight	Alit/Alighted	Alit/Alighted
3	Arise	Arose	Arisen
4	Awake	Awoke	Awoken
5	Be	Was/Were	Been
6	Bear	Bore	Born/Borne
7	Beat	Beat	Beaten
8	Become	Became	Become
9	Begin	Began	Begun
10	Behold	Beheld	Beheld

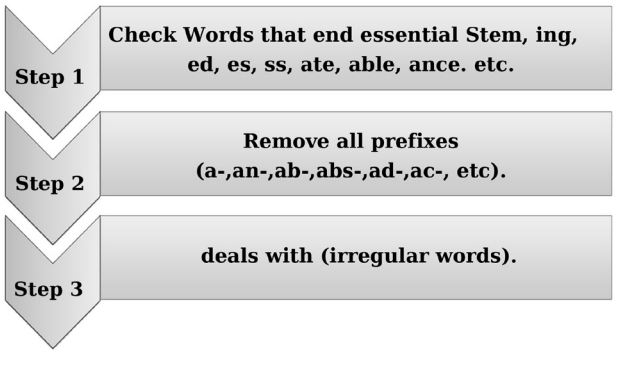

Fig. 6. Ameliorated Porter stemming (APSA)

The first sub-step for words that end with suffixes, which are actually basic words, cannot remove suffixes from these words, it affects the meaning of the word and becomes a non-real word. A list of approximately 753 words has been collected, the importance of the Porter algorithm dealing with these words and returning to the source. The second sub-step is to remove all prefixes. About 160 words have been collected, and they are considered almost all words. The third and final sub-step of the enhancement is that the Porter stemmer deals with irregular words, where the change of the spelling of these words due to the change of time or the case that it represents will make it hard to determine its similarity (about 953 words).

7. Discussion of experimental results

Porter stemmer is approved as the best algorithm for huge documents and this is considered as one of the advantages of the algorithm.

The experiment results show that the Porter stemmer algorithm produces some words having no meaning, while all the words produced from the proposed Porter stemming algorithm have a meaning in English language and morphological analysis in the database even if the document was huge.

The drawback is that the database may not contain all the English words or stems so it cannot give a 100 % accurate result.

For a future scheme, a database can be built containing all English words. In addition, work will be done on other languages and use in a wide range of fields.

To verify the effectiveness of the proposed method, in this section, we perform a test on a set of theses in the pre-processing stage. The system works in three steps:

– the first step is tokenization. We get a good result as shown in Fig. 8. We get a document free of tokens;

– the second step is to remove stopwords. Stopwords are removed or excluded from the specified text so that more emphasis can be placed on those words that define the meaning of the text as shown in Fig. 9. When you remove stopwords, the size of the data set decreases and the time to train the model also decreases. Removing stopwords can help improve performance as there are only a few meaningful words left. This can increase the accuracy of classification;

– the third step is the stemming algorithm. When applying the Porter's algorithm, we observe three defects. First, some of the words suffer from the problem of deleting the suffix of the word, where the word loses the meaning. Second, it does not deal with prefixes, the Porter algorithm cannot remove the prefix. Third, the Porter stemming algorithm does not deal with irregular words, which causes a problem with the meaning and spelling of words.

The results were used for evaluating the performance of the proposed model. The model has been developed using the Matlab programming language, and the tests have been conducted under the environment of the Windows-10 operating system, laptop computer processor: Lenovo laptop, CORE i7, Ram 8 GB, CPU 1.80 GHz. The text below was randomly chosen in corpus to test the effectiveness of the improved algorithm:

("Semantic similarity measures play an important role in the extraction of semantic relations. Semantic similarity measures are widely used in Natural Language Processing (NLP) and Information Retrieval (IR). The work proposed here uses web-based metrics to compute the semantic similarity between words or terms and also compares with the state-of-the-art. For a computer to decide the semantic similarity, it should understand the semantics of the words. The computer, being a syntactic machine, cannot understand the semantics. So an attempt is always made to represent the semantics as syntax. There are various methods proposed to find the semantic similarity between words. Some of these methods have used the precompiled databases like WordNet, Brown, Corpus. Some are based on Web Search Engine. The approach presented here is altogether different from these methods").

Fig. 7 illustrates the interface of the preprocessing phase.

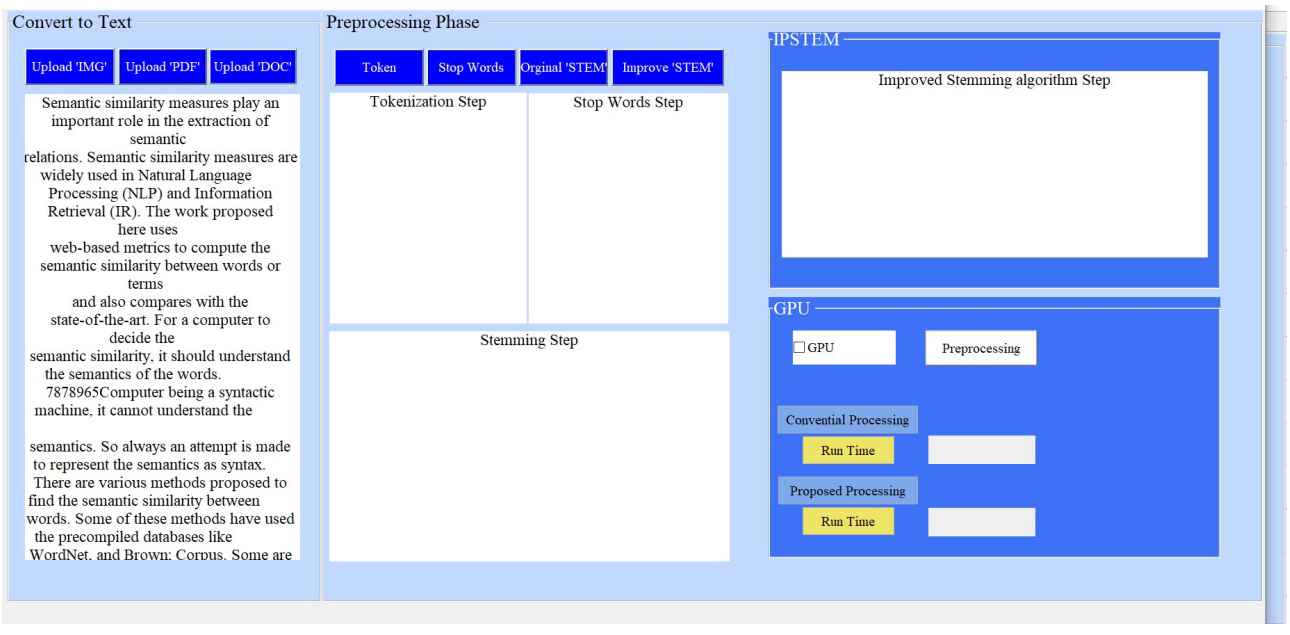

Fig. 7. Illustration of the preprocessing phase

Fig. 7 shows the preprocessing phase. We choose the document by clicking upload. Documents can be uploaded in different formats (PDF, IMG or DOC), documents are uploaded to be processed in several steps.

7. 1. Tokenization step

At the first step, the document enters the tokenization step to remove the token, as shown in Fig. 8. Click on the Tokenization button starts document processing and TXT without the tokens appears.

7. 2. Stopword removal step

After the step of removing the token, we move to the next step in the pre-processing of data, which is to remove stop-words, which are the most common words in the language that are excluded and not indexed in order to improve the search as shown in Fig. 9.

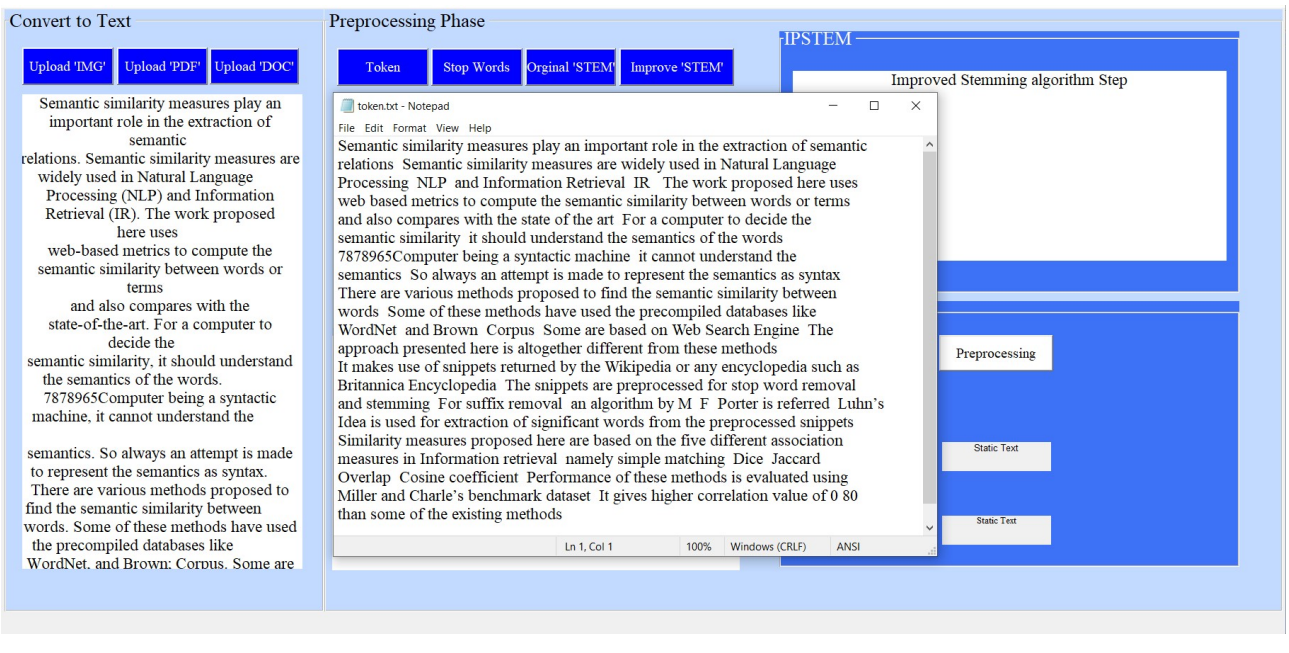

Fig. 8. Illustration of the tokenization step

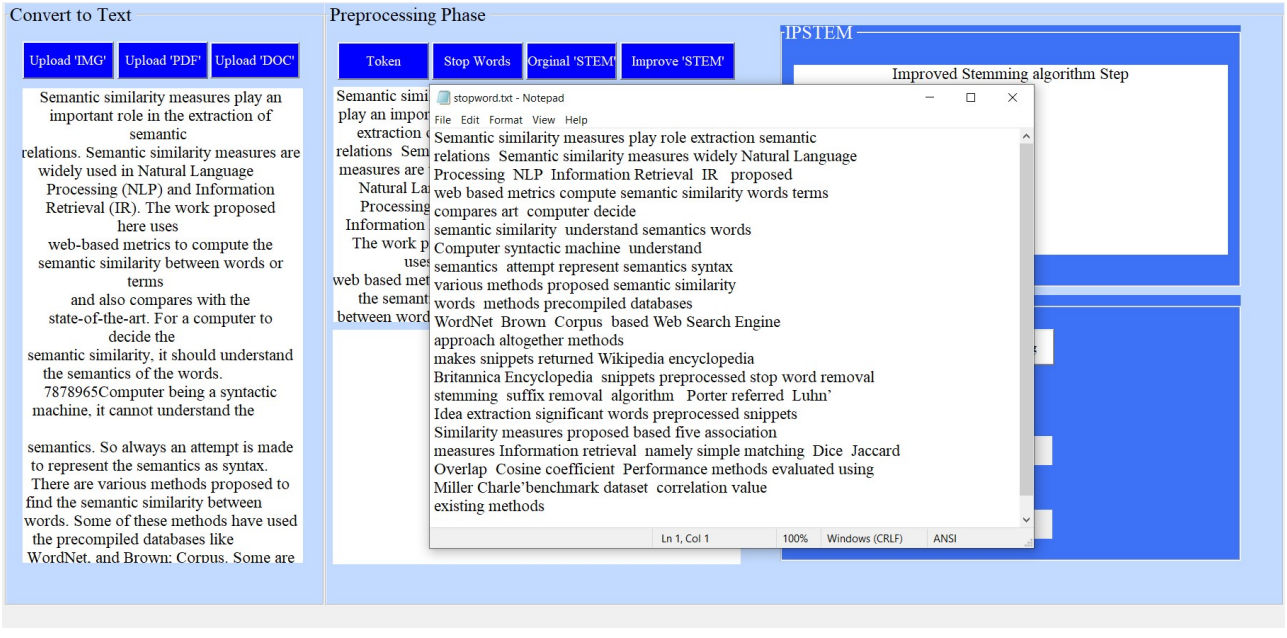

Fig. 9. Illustration of the stopword removal step

7. 3. Stemming algorithm step

In the next step in the pre-processing of data, applying the Porter stemming step we observe the results, after applying the Porter stemming algorithm, most words lost the meaning, and hence we face a major problem as shown in Fig. 10.

Fig. 10 represents the original Porter stemming algorithm. We note the deletion of the endings, where the word loses the meaning. I also cannot calculate the similarity between the two words in case I have a database because it does not apply to the word. This result leads to the need to improve the Porter stemmer, as shown in Fig. 11.

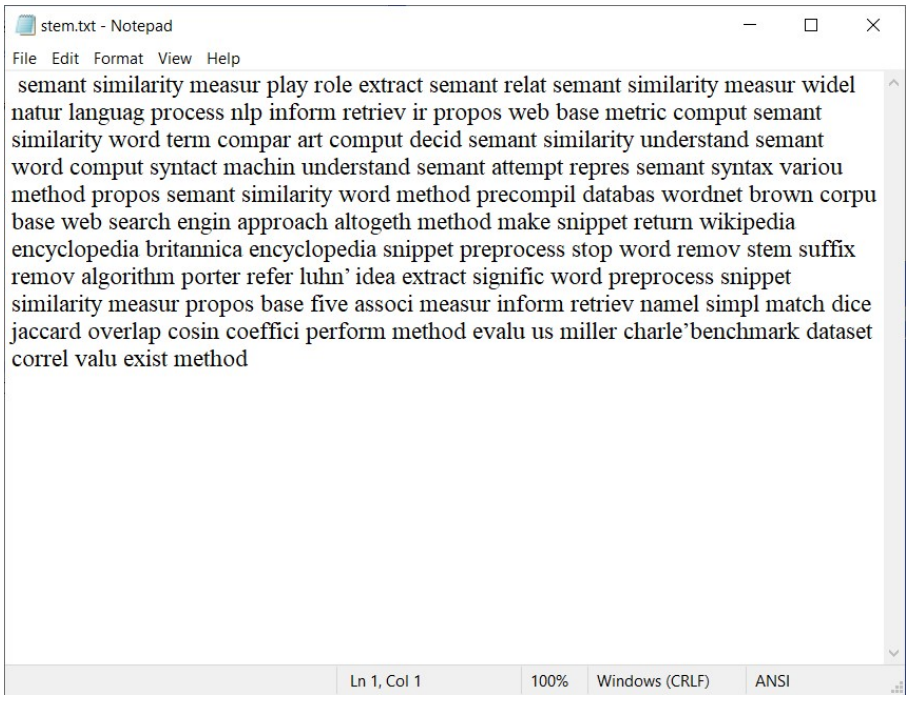

Fig. 10. Illustration of the Porter stemming algorithm result

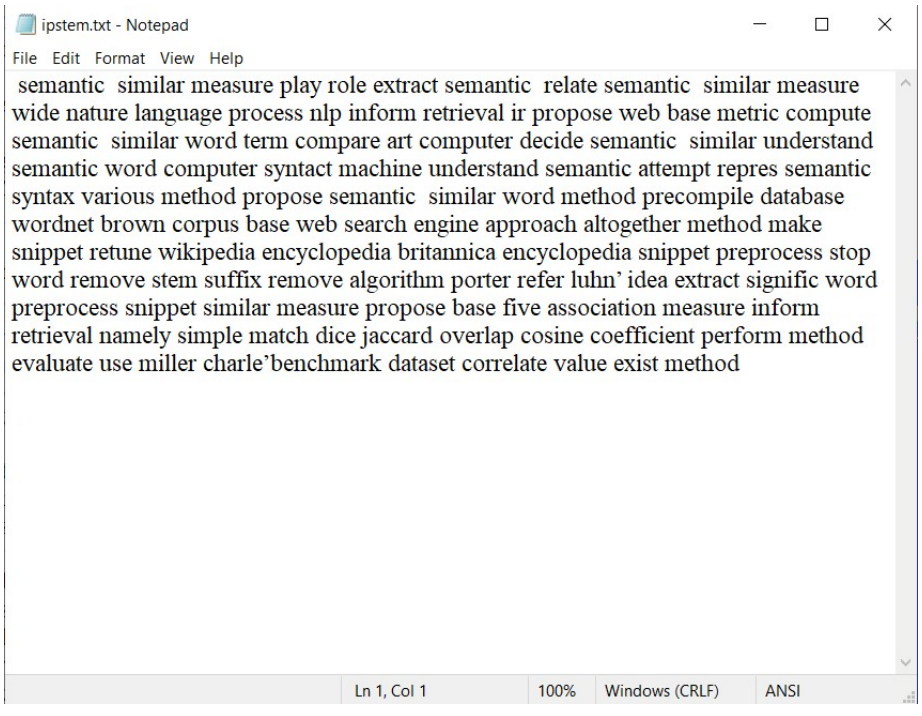

Fig. 11. Result of the ameliorated Porter stemmer

We can see the difference between Fig. 10 and 11, where Fig. 13 illustrates the Porter algorithm after modification by adding 3 new rules and applying to the algorithm. The produced words have meaning from the proposed algorithm and hence we get an accurate result.

8. Conclusions

1. Stemming algorithm is based on the first rule. We obtained real words without deleting their endings. This helps us keep the meaning of the word, leading to accurate and good results.

2. Also, the stemming algorithm through the second rule removes prefixes if present in the features according to the second rule of the improved stemming algorithm rules.

3. The third rule, the stemming algorithm deals with irregular words normally and does not deal with in the past simple tense.

References

1. Seddiqui, H., Maruf, A. A. M., Chy, A. N. (2016). Recursive Suffix Stripping to Augment Bangla Stemmer. ICAICT-2016-Paper. Available at: http://www.ciu.edu.bd/icaict2016/publications/ICAICT-2016-Paper%20(50).pdf
2. Shah, F. P., Patel, V. (2016). A review on feature selection and feature extraction for text classification. 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). doi: https://doi.org/10.1109/wispnet.2016.7566545
3. Saeed, A. M., Rashid, T. A., Mustafa, A. M., Agha, R. A. A.-R., Shamsaldin, A. S., Al-Salihi, N. K. (2018). An evaluation of Reber stemmer with longest match stemmer technique in Kurdish Sorani text classification. Iran Journal of Computer Science, 1 (2), 99–107. doi: https://doi.org/10.1007/s42044-018-0007-4
4. Agbele, K., Adesina, A., Azeez, N., Abidoye, A. (2012). Context-Aware Stemming algorithm for semantically related root words. African Journal of Computing & ICT, 5 (4), 33–42.
5. Akkus, B. K., Cakici, R. (2013). Categorization of Turkish News Documents with Morphological Analysis. 51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop. Sofia, 1–8. Available at: https://www.aclweb.org/anthology/P13-3001.pdf
6. Kumar, R., Mansotra, V. (2016). Applications of stemming algorithms in information retrieval-a review. International Journal of Advanced Research in Computer Science and Software Engineering, 6 (2), 418–423.
7. Biba, M., Gjati, E. (2014). Boosting Text Classification through Stemming of Composite Words. Recent Advances in Intelligent Informatics, 185–194. doi: https://doi.org/10.1007/978-3-319-01778-5_19
8. Farrar, D., Huffman Hayes, J. (2019). A Comparison of Stemming Techniques in Tracing. 2019 IEEE/ACM 10th International Symposium on Software and Systems Traceability (SST). doi: https://doi.org/10.1109/sst.2019.00017
9. Al-Sharhan, S., Al-Hunaiyyan, A., Alhajri, R., Al-Huwail, N. (2019). Utilization of Learning Management System (LMS) Among Instructors and Students. Advances in Electronics Engineering, 15–23. doi: https://doi.org/10.1007/978-981-15-1289-6_2
10. Joshi, A., Thomas, N., Dabhade, M. (2016). Modified Porter Stemming Algorithm. International Journal of Computer Science and Information Technologies, 7 (1), 266–269.