IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 3, Ver. II (May.-June. 2017), PP 44-50
www.iosrjournals.org

Efficient Algorithms for Preprocessing and Stemming of Tweets in a Sentiment Analysis System

Dr. Hussein K. Al-Khafaji, Areej Tarief Habeeb
[1](*Communication engineering department / Al-Rafidain University College, Iraq)*
[2](*Vice President for Scientific Affairs office/ University of Technology, Iraq)*

Abstract : *The preprocessing step approximately consumes 85% of the time and efforts of overall time and efforts of the Knowledge Discovery in Database, (KDD). Sentiments analysis, as a new trend in KDD and data mining, requires many preprocessing steps such as tokenization, stop words removing, and stemming. These steps play, in addition to their preparation role, the data reduction role by excluding worthless data and preserving significant data. This paper presents the design and implementation of a system for English tweets segmentation, cleaning, stop words removing, and stemming. This system implemented as MS-SQL Server stored procedures to be part of a tightly coupled sentiments mining system. Many experiments accomplished to prove the validity and efficiency of the system using different sizes data sets arranged from 250000 to 1000000 tweets and it accomplished the data reduction process to achieve considerable size reduction with preservation of significant data set's attributes. The system exhibited linear behavior according to the data size growth.*

Keywords: *Pre-processing, sentiment analysis, Stemming, Stop words, Tokenization, Twitter.*

I. Introduction

Social platforms have become an important site for conversations throughout the world. People around the world are using the Internet to communicate and express themselves freely, due to the ease of use and apparatus availability. Social media shows the people interact with events and opinions expressions about everything going on in the world.

Twitter is one of the most famous social media networks; it allows users to post tweets, messages of up to 140 characters, on its social network. Twitter usage is growing rapidly. The company reports over 100 million active users worldwide, together sending over 340 million tweets each day (Twitter, March 21, 2012). [1]. It is one of the ten most-visited websites. As of March 31, 2016, Twitter has 310 million monthly active users. [2]

Sentiment Analysis is "the computational study of people's opinions, attitudes and emotions toward an entity. The entity can represent individuals, events or topics. These topics are most likely covered by reviews." [3] Sentiment Analysis recognize the sentiment that a specific text express then analysis it in the form of negative and positive.

Twitter as one of the most visited and used social network, it is a very important resource for data about people interest. To obtain useful information about a particular topic, tweets related to this topic could analyzed. There are many techniques for Sentiment analysis, before applying any one of them, data transformed into structured form, a form that raises the level of performance of the analysis process and consumes less time and less storage. We get this formula through several steps known as preprocessing.

Preprocessing is using techniques for preparing data for the analysis process. It includes several steps, each step produces data ready for the next step until transforming process done, and the data is in the best formula for analyzing.

In this paper, the three crucial steps of preprocessing are discussed, namely Tokenization, Stop word Removal, and Stemming.

Tokenization is the process of transforming a string of words into valuable terms. In any information retrieval model, tokenization is a significant action. It simply detaches the words, numbers, symbols, and any characters, from a text. These words, numbers, symbols and other characters distinguished by tokenization called tokens. [4].

Stop word removal removes words that are useless in information retrieval which known as stop words. These words have no value (positive or negative) in a sentiment analysis system. Therefore, they must be removed from the data set. For example stop words include "the", "as", "of", "and", "or", "to", etc. Stop word removing is substantial in the preprocessing, it has some advantages like reducing the size of stored data set and it improves the overall efficiency and effectiveness of the analysis system. [4]. The proposed system uses a list of stop words obtained from Onix Text Retrieval Toolkit website [5].

Stemming is "a technique to detect different derivations of morphological variants of words in order to reduce them to one particular root called stem" [6]. The most primary form of a word called a stem. Obtaining relevant information from a text written in natural language is a complex process. Natural Languages described

by diverse morphological variants of words, this causes mismatch vocabulary. For text analysis systems, every word should be represented by its stem rather than by the original form of it in the text. [6] As a simple example, consider searching for a document entitled "How to drive a car". If the user wrote, “driving” in the query, there will be no match with the title. However, if the words in the query stemmed, so that “driving” becomes “drive”, then retrieval will be successful. Stemming algorithms are variant, as of now one of the most prevalent stemming algorithm was introduced by Porter (1980). "The Porter stemmer is a process for removing the commoner inflectional and morphological endings from words". [7] The original algorithm has been modified and enhanced many times. The suffixes in the English language (approximately 1200) are often consist of a combination of smaller and simpler suffixes. The basic of Porter algorithm depends on this fact. Porter algorithm includes about 60 rules and is very easy to understand. It has five steps, in each step, a set of rules are applied. When one of the rules passes the conditions, the suffix removed consequently, and then the next step performed. The algorithm returns the stem obtained from the end of the fifth step. [8] The first step treats inflectional suffixes, the next three steps treat derivational suffixes, and the final one is the recoding step.

Porter uses a measure, which is the number of consonant-vowel- consonant string remaining after removal of a suffix. This is a form of a typical rule:

(m>0) *FULNESS *FUL

This rules means that the suffix *FULNESS must be replaced by the suffix *FUL if, and only if, the produced stem has a measure (m) larger than zero.

This algorithm has an iterative manner; it removes a long multi-component suffix in steps. For example, this rule:

(m>0) *FUL null

It means that the suffix *FUL must be replaced by null, which means removing the suffix if, and only if, the produced stem has a measure (m) larger than zero. This rule executes after executing the previous rule which involves the suffix *FULNESS, so the word HOPEFULNESS will be stemmed to HOPEFUL at first, and then to HOPE in the second iteration. [9] [10]

The aim of this research is presenting design and implementation of algorithms to preprocess tweets databases to be ready for mining process such as sentiments mining. These algorithms include an algorithm for segmentation, cleaning, stop words removing, and stemming. The design, implementation, and experiment results illustrated in the next subsections. The proposed system uses data set tweets obtained from large dataset belongs to the University of Michigan Sentiment Analysis competition on Kaggle [11].

II. Proposed Tweets Preprocessing System

For better analysis results, a preprocessing stage should be efficient. In this article, the proposed system involves three steps. These steps save analyzing time and storage space, especially for a huge set of data, in addition to data reduction; it increases the accuracy of the analysis system result. The three steps of the preprocessing stage are Tokenization, Stop word removing, and stemming which shown in fig. (1).

Fig.1 preprocessing steps

1.1. Tokenization

In order to transform a tweet into tokens (discrete words) the tweet passes through two phases: Word segmentation and Cleaning.

1.1.1. As it has mentioned previously, word segmentation: is the process of separating the statement(s) of written language to its words, which compose the sentence(s) structure. The proposed system, analyzes tweets written in English. Tweets are short sentences, so the proposed algorithm divides the sentence into

words and symbols (which are separated by spaces), and stores each word or symbol in a separate row in a table. Fig. (2) shows the algorithm of this step. This algorithm fetches a tweet from tweets table. In step#101, next space function determines the next space in the tweet. Step#104, trim substring function trims the substring occupies the position from P to I as a token, which will be stored in the database. This process will continue until the end of the tweet T. For example if the tweet was " It will be a very hot summer, I think @The-knight", the tokens are "It", "will", "be", "a", "very", "hot", "summer,", "I", "think", "@The-knight" . We notice that the word "summer" and the comma "," are considered together as a single token, because they are not separated by a space. Likewise the phrase "@The-knight". This case will treated in the next phase.

```
Word segmentation algorithm
Input   Tweets_table
                                Output Tokens_table
100   while (more tweets exist in tweets table) do
101   {   T= next tweet; P=1;
102       while not end of T
103       {     I= next_space (T); // determines the position of next space in T
104             token= trim_substring (T, P, I, rest_string);
105             insert into tokens_table (tweet number, token);
106             T= rest_string ;}
107   }
```

Fig.2 word segmentation algorithm

1.1.2. Cleaning: the previous phase produced a collection of words, symbols, punctuation marks, and numbers. Analysis system requires meaningful words that refer to a value (positive or negative), therefore it is necessary to remove unvalued words and symbols from the stored words obtained from the previous phase. Some of these words are expressions that begin with the signs @ and #. They do not have a value, because they are used for special purposes in twitter. For example the @ sign is used to call usernames in Tweets, such as "Good morning @twitter!". A twitter user can use a nominated user @username to mention another user in Tweets, send him a message, or link to his profile. In addition, in this phase the single letters, numbers, and punctuation marks will be removed. This phase will reduce the storage used for the data set and keeps only the effective data that will be used for the sentiment analysis system. Removing the single letters and the words of two letters except the word "no", because it changes the meaning of the sentences, which affects the analysis process. Fig. (3) shows the algorithm of this step. This algorithm fetches a token from tokens table. If the token consisted of one or more symbols as a prefix, steps 205–206 will remove these symbols. The trimleft function trims the first letter from the token. The letter is removed if it is a member in the symbol set presented in step#200. If the token consisted of one or more symbols as a suffix, the steps numbered 209-210 will remove these symbols. The trimright function trims the last letter from the token. The letter is removed if it is a member in the symbol set .Then the token will remain as a word containing letters only. The two symbols (#) and (@) are not removed because the entry of tokens that begins with them will be removed from tokens table, in addition to the tokens that consist one or two letters, or if the token consists of symbols only with no letters. In these cases the entry of tokens will be removed too from tokens table, this is done in step numbered 212. For the previous example, the tokens will be cleaned to give this result: "It", "will", "be", "very", "hot", "summer,", "think". The word "summer," considered as a token after removing the comma. The phrase "@The-knight" is removed from tokens table, because it refers to the username "The-knight" and it will not has a value as positive or negative in the analysis process. The letters "a" and "I" are removed because they are single letters.

```
Cleaning algorithm
Input      Tokens _table
Output   updated   Tokens_table
200 symbols = {,,.,;,:,?,~,`,+,=,%,!,$,^,&,*,",-,_,(,),{,},[,],|,\,/}
201  while (more tokens exist in tokens table) do
202 {T= next token;
203     F= the first letter of T;
204      while F in symbols do // remove symbols from the beginning of token
205      {trimleft (T, F, T);
206        F= the first letter in T ;}
207     E= the last letter of T;
208     while E in symbols do // remove symbols from the end of token
209      {trimright (T, F, T);
210        E=the last letter in T ;}
211     concatenation (T, E, T);
212     if T starts with # or @ or the length of T<=2   then delete T from tokens table}
```

Fig.3 cleaning algorithm

1.2. Stop word removing

Until now, the data stored in the table is large data set, and it contains many words that is not useful for the analyzing system which known as stop words. The stop words of English language are stored in a table in this step, items in the tokens table compared with each word in the stop word table in order to delete the stop words from tokens table for each tweet. Fig. (4) shows the algorithm of stop word removing. The same previous example will give this result: the remained tokens are “hot”, “summer”, “think”. The words “It”, “will”, “be”, “very”, are removed as stop words. The method of matching in step# 303 is neglected and not elucidated in this paper.

```
Stop words removing algorithm
Input    Tokens _table
Output updated tokens_table
300   while (more tokens exist in tokens table) do
301 {T= next token;
302       if T in stop_words_table
303        then delete the entry of T from tokens table}
```

Fig.4 Stop words removing algorithm

1.3. Stemming

Now there is a data set that contains only words that have meaning and values, but the amount of data still large. In addition, there are many words with a corresponding meaning. Stemming is to find out the root/stem of a word. For example, the words observe, observes, observer, and observation all could be stemmed to the word "Observe". The purpose of this step is to remove various suffixes, to have exactly matching stems, to save memory space and time. The proposed system uses Porter algorithm for stemming.

III. Experimental results

In this section, the experimental results of the proposed preprocessing system discussed. The experiments are accomplished using data set containing one million tweets. Table.1 shows a sample of the data set.

Table .1 sample of the data set

Tweet no.	Tweets
1	"very nice film I love it
2	"nice time I love this film
3	"I loved it very much nice film
4	"not nice film, so long, it should be shorter
5	"long film but I love the hero , it is nice

1.4. Tokenization

Executing the tokenizing step on the data set gives the result with 5372158 tokens stored in tokens table. Table.2 shows a sample of the data after the word segmentation and word cleaning. According to table.2, the tokens (very, nice, film, and love) belong to the tweet no.1 and so for the next tweets.

Table.2 sample of the tokens (segmented and cleaned words)

Tweet no.	Token
1	Very
1	Nice
1	Film
1	Love
2	Nice
2	Time
2	Love
2	This
2	film

1.5. Stop words removing

This step removes stop words that saved already in the tokens table depending on the stop words table. It eliminated the number of tokens to 3162653 tokens. Table.3 shows a sample of the remaining tokens after removing stop words. Stop words table contains 403 words,obtained from Onix Text Retrieval Toolkit website.

Table.3 sample of the tokens (after removing stop words)

Tweet no.	Token
1	Nice
1	Film
1	Love
2	Nice
2	Time
2	Love
2	film

1.6. Stemming

In the proposed system, Porter stemmer is used. This step will stem number of words, For example, the tokens "loved" in the tweet no.3, and "shorter" in the tweet no.4 (shown in table.1) will be stemmed to the words "love" and "short".

1.7. Execution time

The execution time of the proposed system was an hour and 32 minutes for the data set contains one million tweets. It is executed by using a computer with 8.00 GB RAM, Intel(R) Core(TM) i7 CPU 1.87 GHz, and 450 GB hard disk. The algorithms of the preprocessing are implemented using SQL server stored procedure.Table.4 presents the execution time for every step for three data sets with different number of tweets, and the number of tokens obtained as a result after executing tokenizing and stop words removing steps.

Table.4 Execution time and number of tokens of the proposed system steps

The data set size (tweets number)	Tokenizing step		Stop words removing step		Stemming step
	execution time (m:s)	**Number of tokens as a result**	**execution time (m:s)**	**Number of tokens as a result**	**execution time (m:s)**
250000	09:41	1237844	00:23	690640	09:30
500000	20:20	2465297	00:55	1368824	19:07
1000000	45:40	5372182	01:43	3087601	44:39

IV. Conclusion

The problem of preparing data for sentiment analysis system requires several steps. These steps play the role of data reduction in data mining to save mining time and storage space without data loss, especially for a huge data set. Each step reduces the amount of data by removing un-useful items. These items could be symbols, words, or phrases. Then to make the data effective for the analysis system a stemming algorithm used, which decreases the variant words that have similar meaning. Stemming is very critical step for the sentiment analysis system. The preprocessing system is implemented using MS-SQL server, and this may be a reason for its relative slowness.

SQL is not suitable for such applications, but we insisted on using it to design a complete DBMS-embedded miner, however a plan is designed to implement the system using data segmentation and multi-threaded to increase the time efficiency of the system.

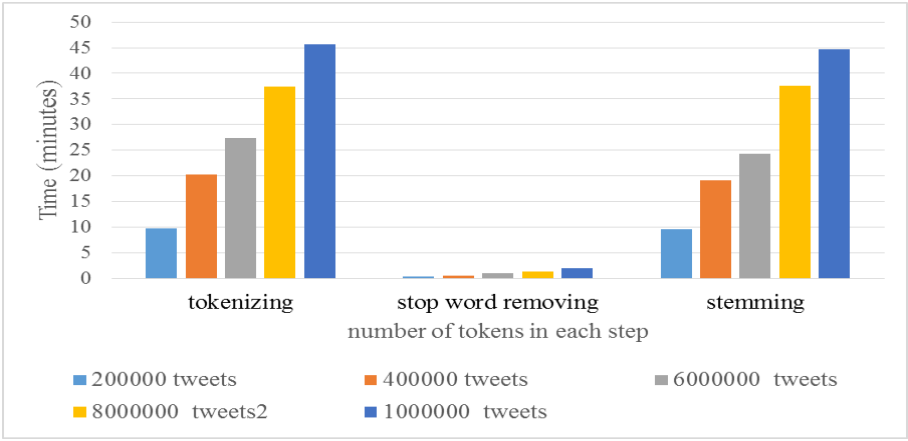

Fig.5 Execution time for each step

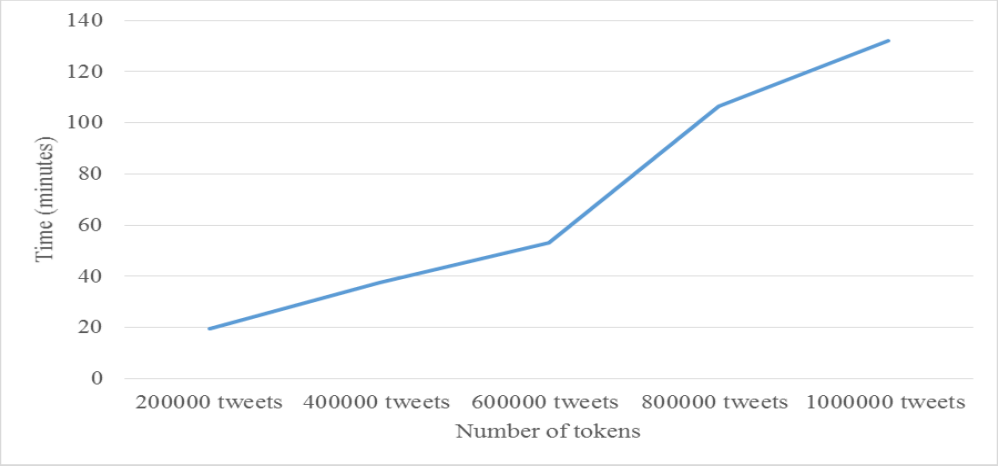

Fig.6 Execution time of five data sets

References

[1] Twitter (@twitter), "Twitter turns six", March 21, 2012.https://blog.twitter.com/2012/twitter-turns-six. Visited at 4/2016.
[2] Twitter usage / company facts. https://about.twitter.com/company. Visited at 7/2016
[3] G.Vinodhini, RM.Chandrasekaran, "Sentiment analysis and opinion mining: a survey", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 6, Page 282, June 2012
[4] Vikram Singh, Balwinder Saini, "An effective tokenization algorithm for information retrieval system", Department of Computer Engineering, National Institute of Technology, Kurukshetra, Haryana, India, David C. Wyld et al. (Eds), 2014.
[5] Onix Text Retrieval Toolkit API Reference.http://www.lextek.com/manuals/onix/stopwords1.html. Visited at 5/2016.
[6] M. Kantrowitz, B. Mohit, and V. Mittal."Stemming and Its Effects on TFIDF Ranking",In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 357–359, Athens, Greece, 2000.
[7] Martin Porter, "The Porter Stemming Algorithm", Jan 2006. http://tartarus.org/~martin/PorterStemmer/index.html . Visited at 5/2016.
[8] Ms. Anjali Ganesh Jivani, "A Comparative Study of Stemming Algorithms", Anjali Ganesh Jivani et al, Int. J. Comp. Tech. Appl., Vol 2 (6), 2011.
[9] Peter Willett, "The Porter stemming algorithm: then and now", White Rose Consortium, White Rose Consortium ePrints Repository, 2006.
[10] M.F.Porter, "An algorithm for suffix stripping", July 1980. http://tartarus.org/~martin/PorterStemmer/def.txt . Visited at 5/2016.
[11] http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22 . Visited at 5/2016.