Component Analysis of a Sentiment Analysis framework on different corpora

Walaa Medhat
School of Electronic Engineering, Canadian International College, Land 6, South of Police Academy, ElTagamoa El-Khames, New Cairo city.
walaa_medhat@cic-cairo.com

Ahmed H. Yousef
Computers & systems Department, Faculty of Engineering, Ain Shams University, Egypt/ 1 Sarayat street, Abbasiya, Cairo.
ahassan@eng.asu.edu.eg

Hoda K. Mohamed
Computers & systems Department, Faculty of Engineering, Ain Shams University, Egypt/ 1 Sarayat street, Abbasiya, Cairo.
hoda.korashy@eng.asu.edu.eg

***Abstract*— Sentiment Analysis (SA) is the computational study of people's opinions about certain topics. With the massive growth of web 2.0 technologies, many sources of data and corpora are available for SA. There are some recent frameworks proposed in this field that can deal with different corpora. This paper presents a component analysis of recently proposed sentiment analysis framework. The framework components are divided to three stages, each of which contains many alternatives. The first stage is the text processing which include "handling negations, removing stopwords, and using selective words of part-of-speech tags". The second stage is the feature extractions which are "unigrams and bigrams". The third stage is the text classification which was done using "Naïve Bayes and Decision Tree" classifiers. It is important to analyze the components of the framework to configure which scenario is better for each corpus used. The analysis is enhanced by applying the framework components on the benchmark corpus movie reviews in addition to the prepared corpora from online social network sites and a review site. The results show that applying all the stages of text processing techniques ultimately decrease the classifiers' training time with no significant penalty in accuracy. The results also show that "Naïve Bayes" gives higher accuracy in case of balanced benchmark corpus while "Decision tree" classifier is better for imbalance data from social network.**

Keywords—Sentiment Analysis; Feature Extraction; Text Processing; Social Network Corpora

I. Introduction

Sentiment Analysis is the computational study of people's opinions, attitudes, and emotions towards individuals, events, or topics covered by reviews or news. The target of SA is to find opinions, identify the sentiments they express, and then classify their polarity. SA at document level aims to classify an opinion document as expressing a positive or negative opinion or sentiment. It considers the whole document a basic information unit (talking about one topic). Reviews are considered documents [1].

SA can be considered also a classification process. Sentiment Classification (SC) is the task of classifying text to represent a positive or negative sentiment. SC is usually formulated as a two-class classification problem; positive and negative. Training and testing data used are usually product reviews. Most research papers do not use the neutral class, which makes the classification problem considerably easier, but some have used the neutral class like [2]. SC is essentially a text classification problem. The traditional text classification mainly classifies documents of different topics, e.g., politics, sciences, or sports. Sentiment or opinion lexicon that indicate positive or negative opinions are more important to SC, e.g., great, excellent, amazing, horrible, bad, worst, etc. Since it is a text classification problem, any existing supervised learning method can be applied, i.e., Naïve Bayes (NB) classifier.

The problem of classifying the benchmark movie reviews into two classes, positive and negative was first tackled by [3]. It was shown that using unigrams in classification gives the highest accuracy with NB. Many researches tend to find more effective features to improve the performance of classification. There are many effective features that affect the classification process such as sentiment shifters. These are the words that reverse the sentiment like the use of negation i.e. "not". Sentiment Shifters were discussed in [4, 5]. Other previous researches used Part-of-Speech (POS) tagging and syntactic dependencies as features [6-9].

There are many other corpora than movie reviews that can be used in SA. Some of them are real data that can be prepared for SA as proposed by [10]. They have proposed a framework for preparing and using corpora from social network sites and a review site for SA task. The framework consisted of three phases. The first phase is the preprocessing and cleaning of data collected, then data annotation. The second phase is applying various text processing techniques including "replacing the negation words and the following negated words with the antonyms of the negated words, removing stopwords, and using selective words of POS tags (adjectives and verbs) "on the prepared corpora. The third phase is text classification using "Naïve Bayes and Decision Tree" classifiers and two different features, "unigrams and bigrams".

The other frameworks proposed in the literature take nearly the same steps of preprocessing and cleaning and most of them apply removing stopwords and POS tagging as in [11,12]. The selected framework has added a new step for handling negation which can be interested to be investigated. The review corpus and the social network corpora were downloaded from three sites IMDB, Facebook, and Twitter respectively. Their prepared corpora used in the framework are publicly available (IMDB[1], Facebook[2], Twitter[3]).

---

[1] http://goo.gl/PSpqg4
[2] http://goo.gl/zxam8w
[3] http://goo.gl/tv53aR

This paper presents a component analysis for the framework proposed in [10]. We have applied the feature extraction methods and the classification process after each stage of the text processing techniques proposed. The tests were made on the benchmark corpus of movie reviews in addition to the corpora prepared in the framework. The benchmark movie reviews corpus has been established as a popular benchmark dataset for SA and is publicly available[4]. There are many previous works have tackled the classical two-class classification problem of text [3, 13, 14]. However, there are still unsolved problems. One of these problems is the problem of handling negations. The framework has proposed a new method of handling negations by replacing the negation words and the following negated words with the antonyms of the negated words.

The main target of the component analysis is to configure which scenario can give the best performance for each corpus. The best combination of the text processing technique, the feature extraction method and the classifier used that gives the highest accuracy and the least training time.

The paper is organized as follows; section 2 presents the methodology. Results and Discussion are presented in section 3. Section 4 presents the conclusion and future work.

II. Methodology

The aim of our work is to analyze the components of the framework which are divided into three stages. The first stage is the text processing which include "replacing the negation words and the following negated words with the antonyms of the negated words, removing stopwords, and using selective words of POS tags (adjectives and verbs)". The second stage is the feature extractions which are "unigrams and bigrams". The third stage is the text classification which was done using "Naïve Bayes and Decision Tree" classifiers. The following subsections discuss the corpora, the components description and the design of the framework.

A. Corpora

We have tested four different corpora. The first corpus is the benchmark movie reviews. This is a corpus of classified movie reviews which contains 2000 movie reviews: 1000 positive and 1000 negative. The reviews were originally collected from the Internet Movie Database (IMDB[5]) review site. Their classification as positive or negative is automatically extracted from the ratings, as specified by the original reviewer [3].

The other corpora were data downloaded from two online social networks (OSN) Twitter and Facebook and the review site (IMDB). The data was on the same topic (a single movie) which is a related topic to the benchmark corpus [10]. The reviews from IMDB were 375 positive and 113 negative. The comments from Facebook were 920 positive and 201 negative. The tweets from twitter were 245 positive and 21 negative.

[4] http://www.cs.cornell.edu/people/pabo/movie-review-data
[5] http://www.imdb.com/

B. Text Processing techniques

There are three text processing techniques used in the tests presented here in some details.

1) Replacing Negations with Antonyms

Negations are words that reverse the sentiment orientation of a sentence. For example consider the sentence "This movie is good" versus "This movie is not good". In the first one the word "good" is a positive term, so the sentence is positive. When "not" is applied to the clause; the word "good" is being used in a negative context, so the sentence is negative [4]. There are many models for handling negations were proposed. A good survey of negation modeling is found in [15].

The basic approach of handling negation was to add artificial words: i.e. if a word "x" is preceded by a negation word, then rather than considering this as an occurrence of the feature "x", a new feature "NOT x" is created [3]. The approach of handling negation proposed in the framework was different from the literature. It simply replaced the negation words and the following negated words with the unambiguous antonyms of the negated words. It suggested that negated words can come after three shifters (not, don't, can't). Whenever one of the shifters appears in the sentence, the word after them and the shifter are replaced with its unambiguous antonym. Unambiguous means if the word has more than one antonym, it will not be replaced. The antonyms were retrieved from Wordnet [16].

2) Removing Stopwords

Stopwords are common words that generally do not contribute to the meaning of a sentence, specifically for the purposes of information retrieval and natural language processing. The common English words that don't affect the meaning of a sentence are like "a", "the", "of"…. Removing stopwords will reduce the corpus size without losing important information.

3) Part-of-speech Tagging

POS tagging is the process of converting a sentence, in the form of a list of words, into a list of tuples, where each tuple is of the form (word, tag). The tag is a POS tag and signifies whether the word is a noun, adjective, verb, …etc [17]. A Classifier based tagger was used to tag the different corpora. This tagger uses the NB classifier which is trained on the Penn Treebank tagged corpus [18]. A good survey of using POS can be found in [19]. It was pointed out in this survey that adjectives are natural indicators of sentiment as proposed by [3]. SA on a corpus containing adjectives only can give good results. There have been some targeted comparisons of the effectiveness of other POS tags including verbs, adverbs and nouns [20-22]. Verbs are chosen as they are good indicator of sentiments too i.e. "love, or hate". The different corpora were tagged then some selective POS tags were considered in the classification. The selective POS were adjectives (base form, comparative and superlative), and verbs (base form, past tense, gerund or present participle, past participle, non-3rd person singular present, and 3rd person singular present).

C. Feature Extraction

There are two Features extraction methods used in the test:

-Unigram: treats the documents as bag-of-words (BOWs) which constructs a word presence feature set from all the words of an instance.

-Bigram: is the same as unigram but finds pair of words.

These two methods are widely used in the SA field [23]. These tests can be extended by extracting more features i.e. n-grams.

D. Text Classification

There are two supervised learning classifiers used; Naïve Bayes (NB) [24] and Decision tree (DT) [25]. There are many other kinds of supervised classifiers in the literature [23]. The two chosen classifiers represent two different families of classifiers. NB is one of the probabilistic classifiers. It is the simplest and most commonly used classifier. DT on the other hand is a hierarchical decomposition of data space and doesn't depend on calculating probability. The two classifiers were conducted with the nltk 2.0 toolkit. There are some parameters passed in to the DT classifier [17].

The parameters are:

-Entropy cutoff: used during the tree refinement process. If the entropy of the probability distribution of label choices in the tree is greater than the entropy_cutoff, then the tree is refined further. But if the entropy is lower than the entropy_cutoff, then tree refinement is halted. Entropy is the uncertainty of the outcome. As entropy approaches 1.0, uncertainty increases and vice versa. Higher values of entropy_cutoff will decrease both accuracy and training time. It was set to '0.8'.

-Depth cutoff: used during refinement to control the depth of the tree. The final decision tree will never be deeper than the depth_cutoff. Decreasing the depth_cutoff will decrease the training time and most likely decrease the accuracy as well. It was set to '5'.

-Support cutoff: controls how many labeled feature sets are required to refine the tree. When the number of labeled feature sets is less than or equal to support_cutoff, refinement stops, at least for that section of the tree. Support_cutoff specifies the minimum number of instances that are required to make a decision about a feature. It was set to '30'.

E. Design of the SA framework

Fig. 1 shows the design of the SA framework including the text processing and the classification on a prepared corpus. In order to analyze the components of the framework, the classification was performed after applying each stage of text processing technique as follows:

-Path A: testing on original data without any text processing.

-Path B: testing on original data after replacing the negation words and the following negated words with the antonyms of the negated words.

-Path C: testing on original data after replacing the negation words and the following negated words with the antonyms of the negated words then removing the stopwords. The stopwords couldn't be removed at first as the shifters will be removed and the negation words can't be detected.

-Path D: testing on original data after replacing the negation words and the following negated words with the antonyms of the negated words then, removing the stopwords then, selective POS tagging of adjectives and verbs.

After applying each text processing stage, the corpus size is reduced and a new modified version of the original corpus is produced. The classification was done four times after each path as follows:

-NB with unigrams

-NB with bigrams

-DT with unigrams

-DT with bigrams

These tests were all performed using the Natural Language Toolkit (nltk 2.0) which is implemented inside python 2.6 [17].

III. Results and Discussion

We used a HP pavilion desktop computer of model: p6714me-m. The processor is Intel(R) core (TM) i5-2300 CPU @ 2.80 GHZ; RAM is 4GB; and 64-bit operating system. We have calculated the training time using a build-in function written with python code which calculates the processing time in seconds [3].

We have made many experiments to test the components of the SA framework mentioned in the previous section. We have made the tests after splitting 75% of the total number of the data in each corpus for training and 25% for testing data.

The standard Accuracy and E-measure were used to evaluate the performance for each test. The accuracy is defined as: the ratio of number of correctly classified data to the total number of data. F-measure is computed by: combining the Precision and Recall in the following way:

$$F - measure = \frac{2 \times precision \times recall}{precsion + recall} \qquad (1)$$

where precision is defined as the ratio of number of correctly assigned category C reviews to the total number of reviews classified as category C. Recall is the ratio of correctly assigned category C reviews to the total number of reviews actually in category C. Since F-measure is computed for each category separately, we aggregated the F-measure scores by using the Macro average of the F-measure scores. Macro-average gives each category equal weight.

However, the precision and recall sometimes give zeroes for a certain polarity. This happens with the extremely unbalanced data Twitter when using DT. The precisions and recalls for negative polarity give zeroes. Therefore, the F-measure of Twitter when using DT as shown in Table 1 is for positive polarity only.

Table 1 shows the F-measure of the various tests we have made. We can notice that using NB classifier gives the highest F-measure when applied on benchmark data and gives slightly better F-measure than DT when applied on OSN data and

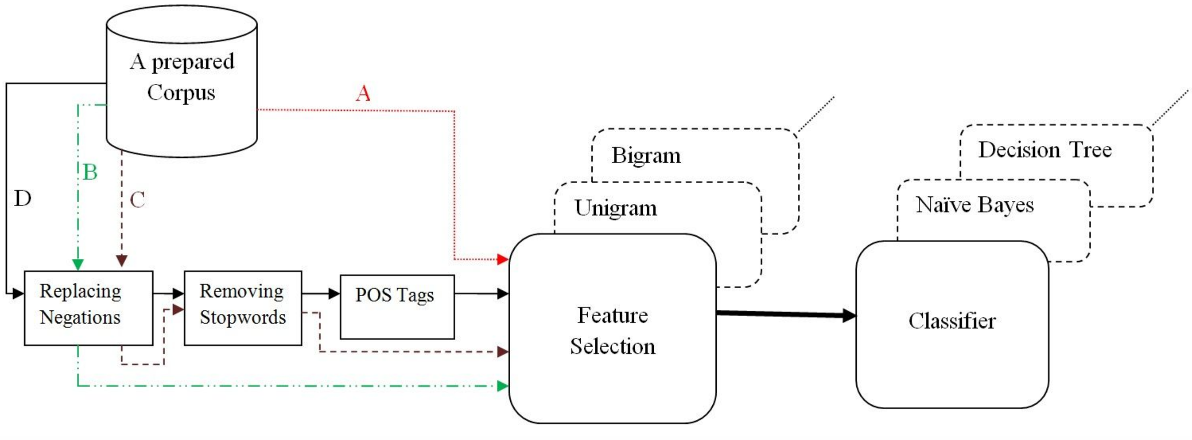

Fig. 1 Design of the SA framework

TABLE I F-MEASURE OF SENTIMENT ANALYSIS ON MOVIE REVIEWS, IMDB, FACEBOOK AND TWITTER CORPORA USING NB AND DT CLASSIFIERS WITH UNIGRAM AND BIGRAM AS FEATURES AFTER APPLYING DIFFERENT PATHS OF TEXT PROCESSING TECHNIQUES

Classifier	Feature selection	Processing	F-measure			
			Movie Reviews	*IMDB*	*Facebook*	*Twitter*
Naïve Bayes	Unigram	*Path A*	0.706	0.661	0.756	0.522
		Path B	0.719	0.730	0.696	0.501
		Path C	0.708	0.771	0.686	0.346
		Path D	0.732	0.726	0.593	0.627
	Bigram	*Path A*	0.811	0.540	0.730	0.397
		Path B	0.817	0.669	0.654	0.402
		Path C	0.768	0.733	0.647	0.239
		Path D	0.794	0.738	0.598	0.535
Decision Tree	Unigram	*Path A*	0.682	0.598	0.536	0.95
		Path B	0.661	0.617	0.489	0.95
		Path C	0.669	0.538	0.507	0.95
		Path D	0.665	0.513	0.507	0.95
	Bigram	*Path A*	0.683	0.598	0.468	0.95
		Path B	0.668	0.617	0.489	0.95
		Path C	0.683	0.538	0.507	0.95
		Path D	0.661	0.513	0.470	0.95

review data except for Twitter corpus which is extremely unbalanced.

The following figures from 2 to 9 show the accuracy and training time for the different corpora after applying each path of text processing.

Fig. 2 shows that NB give higher accuracy than DT in case of testing on benchmark data and that bigrams are better than unigrams when using NB. Path D increase the accuracy in case of using NB with unigrams. The text processing techniques decrease the accuracy with DT classifier slightly but decrease the training time of DT dramatically as shown in Fig. 3. This figure shows the logarithmic graph of the classifiers' training time on Movie Reviews. It shows that DT takes much longer training time than NB. The text processing techniques also decrease the training time of NB classifier.

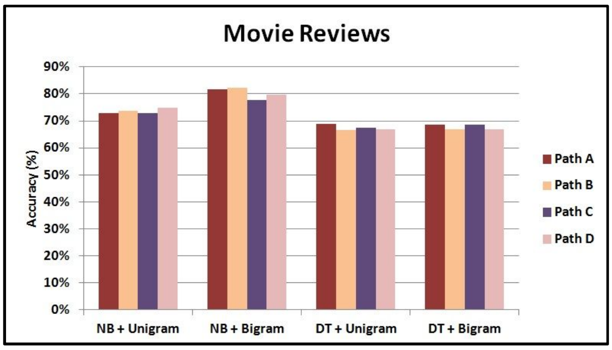

Fig. 2 Classification accuracy of Movie Reviews corpus after applying different paths of text processing

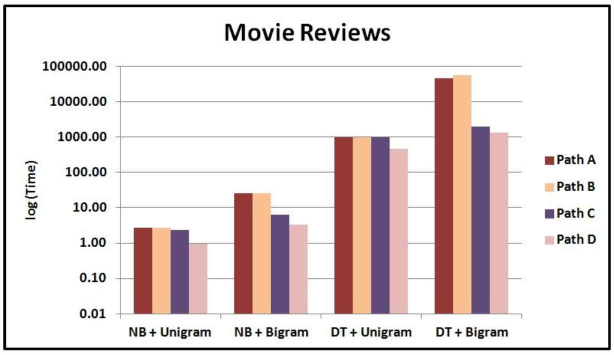

Fig. 3 Classification training time of Movie Reviews corpus after applying different paths of text processing

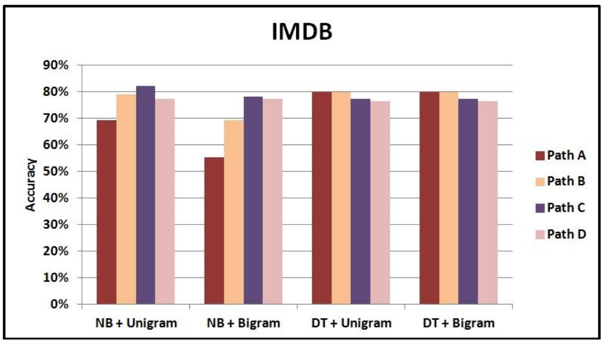

Fig. 4 Classification accuracy of IMDB corpus after applying different paths of text processing

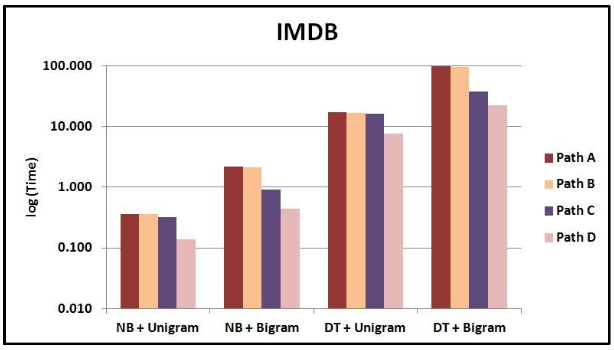

Fig. 5 Classification training time of IMDB corpus after applying different paths of text processing

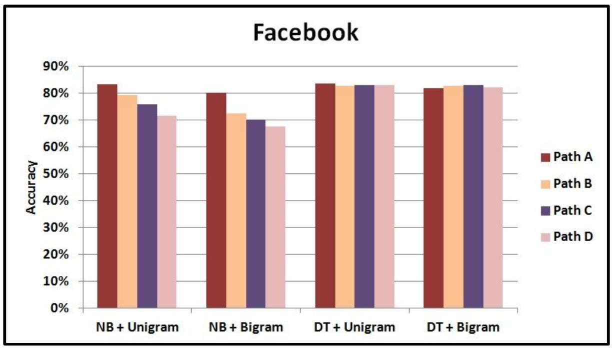

Fig. 6 Classification accuracy of Facebook corpus after applying different paths of text processing

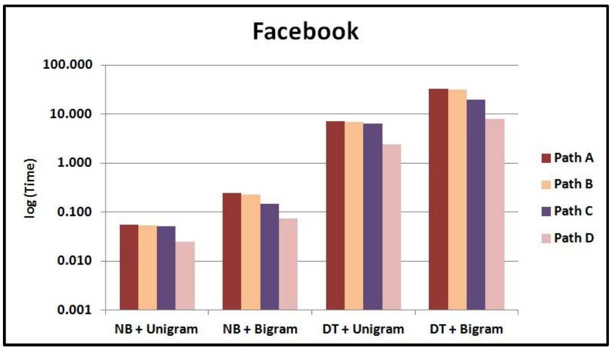

Fig. 7 Classification training time of Facebook corpus after applying different paths of text processing

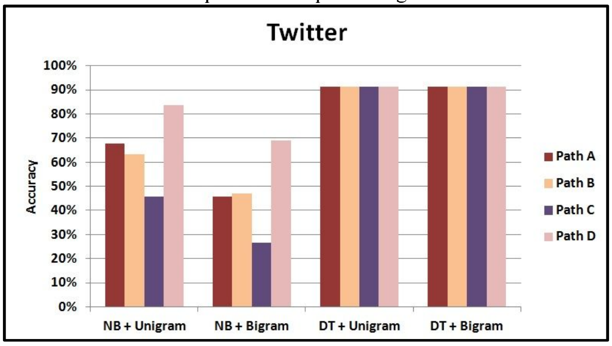

Fig. 8 Classification accuracy of Twitter corpus after applying different paths of text processing

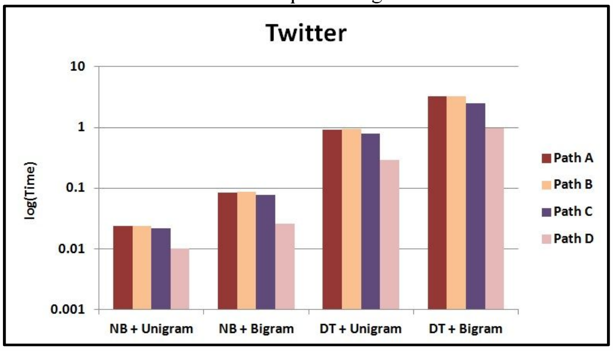

Fig. 9 Classification training time of Twitter corpus after applying different paths of text processing

Fig. 4 shows that DT gives higher accuracy than NB without applying any text processing in case of testing on reviews corpus and that unigrams are better than bigrams when using NB. There is no significant difference between unigrams and bigrams when using DT. The text processing techniques increase the accuracy of NB classifier but don't affect the accuracy of the DT classifier much. The training time of DT is still higher than NB but the difference is not so big like the benchmark corpus. The text processing techniques decrease the training time of both classifiers as shown in Fig. 5.

Fig. 6 shows that DT gives higher accuracy than NB in case of testing on Facebook corpus. Unigrams are slightly better than bigrams when using both classifiers. The text processing techniques decrease the accuracy of NB classifier but don't affect the accuracy of the DT classifier much. The training time of DT is higher than NB and the text processing techniques decrease the training time of both classifiers as shown in Fig. 7.

TABLE II MAXIMUM ACCURACIES, F-MEASURE AND MINIMUM TRAINING TIME ACHIEVED BY EACH CORPUS WITH COMBINATION OF TEXT PROCESSING PATHS, FEATURES AND CLASSIFIERS

<table>
<tr><th></th><th></th><th>Path A</th><th>Path B</th><th>Path C</th><th>Path D</th><th>Unigram</th><th>Bigram</th><th>NB</th><th>DT</th></tr>
<tr><td rowspan="3">Movie Reviews</td><td>Max. Accuracy (82.2 %)</td><td></td><td>√</td><td></td><td></td><td></td><td>√</td><td>√</td><td></td></tr>
<tr><td>Max. F-measure (0.817)</td><td></td><td>√</td><td></td><td></td><td></td><td>√</td><td>√</td><td></td></tr>
<tr><td>Min. Time (0.95 sec)</td><td></td><td></td><td></td><td>√</td><td>√</td><td></td><td>√</td><td></td></tr>
<tr><td rowspan="3">IMDB</td><td>Max. Accuracy (82.11 %)</td><td></td><td></td><td>√</td><td></td><td>√</td><td></td><td>√</td><td></td></tr>
<tr><td>Max. F-measure (0.771)</td><td></td><td></td><td>√</td><td></td><td>√</td><td></td><td>√</td><td></td></tr>
<tr><td>Min. Time (0.14 sec)</td><td></td><td></td><td></td><td>√</td><td>√</td><td></td><td>√</td><td></td></tr>
<tr><td rowspan="3">FB</td><td>Max. Accuracy (83.63 %)</td><td>√</td><td></td><td></td><td></td><td>√</td><td></td><td></td><td>√</td></tr>
<tr><td>Max. F-measure (0.756)</td><td>√</td><td></td><td></td><td></td><td>√</td><td></td><td>√</td><td></td></tr>
<tr><td>Min. Time (0.03 sec)</td><td></td><td></td><td></td><td>√</td><td>√</td><td></td><td>√</td><td></td></tr>
<tr><td rowspan="3">TW</td><td>Max. Accuracy (91.17 %)</td><td colspan="7">√</td><td>√</td></tr>
<tr><td>Max. F-measure (0.95)</td><td colspan="7">√</td><td>√</td></tr>
<tr><td>Min. Time (0.01 sec)</td><td></td><td></td><td></td><td>√</td><td>√</td><td></td><td>√</td><td></td></tr>
</table>

Fig. 8 shows that DT gives much higher accuracy than NB in case of testing on Twitter corpus and that unigrams are better than bigrams when using NB. There is no significant difference between unigrams and bigrams when using DT. Applying all text processing techniques dramatically increase the accuracy of NB classifier but don't affect the accuracy of the DT classifier. The training time of DT is still higher than NB and the text processing techniques decrease the training time of both classifiers as shown in Fig. 9.

Replacing the negation words and the following negated words with the antonyms of the negated words then removing stopwords then tagging the dataset and selecting adjectives and verbs only has reduced the size of the corpus by more than half of its original size. This reduction results in decreasing the classifiers' training time with no significant penalty in accuracy in many cases but sometimes it is better especially with the Twitter data. This proves that adjectives and verbs are very important indicators for sentiments.

In NB tests on benchmark corpus, the accuracy is better when using bigram. This contradicts with what [3] has found as they worked on movie reviews too. They found that using unigrams give better results. But NB is better with unigrams in the OSN tests. In DT tests the accuracy is lower than NB tests on benchmark data which was found in the literature [23]. But it is higher than NB tests on OSN data. The difference between the benchmark data and the OSN data and the reviews as well is that the benchmark is balanced data where the number of positive reviews is equal to the number of negative reviews. The data downloaded from the OSN sites and the review site; were unbalanced as the number of positive class is much bigger than the negative class. DT gives better accuracy than NB for unbalanced data and convergent F-measure. NB calculates the probability on the whole data but DT is more specifically build hierarchy decomposition of data. That is why DT is better for unbalanced data as it is more specific than NB. A good comparison between several machine learning algorithms could be found in [26]. In the literature, DT was targeted for many researches and development to be suitable for imbalance data as in [27-31]. They showed that DT could be suitable for imbalance data.

In these tests we have tried several combinations of classifiers and features along with text processing techniques to configure which path of text processing with which classifier and feature can give the best accuracy for each corpus. Table 2 sums up the best accuracy, the best F-measure and the least training time achieved by each corpus illustrating the combination of text processing path, feature and classifier.

Table 2 shows that DT gives the best accuracy for the OSN data while NB gives the best accuracy for the reviews data. The F-measure of NB is higher in most corpora except for the extremely unbalanced data of Twitter. The text processing techniques decrease the accuracy but decrease the training time as well. Path D always gives the minimum time when using it with NB and unigrams. It also gives the least training time among other paths with any combination of classifiers and features as shown on Fig. 3, 5, 7, and 9.

IV. CONCLUSION AND FUTURE WORK

In this paper we have presented a component analysis of a recently proposed sentiment analysis framework. The paper aims at analyzing the effectiveness of applying each stage of the text processing techniques proposed in the framework, which were replacing the negation words and the following negated words with the antonyms of the negated words, removing stopwords, and using selective words of POS tags (adjectives and verbs). The corpora used for testing were the benchmark corpus movie reviews, reviews downloaded from IMDB site, comments from Facebook and tweets from twitter. The data downloaded are all on the same topic (a single movie). We performed the tests on two well known classifiers (Naïve Bayes, Decision tree) and two different features (unigram, bigram). We have evaluated the performance of the classifiers by the means of accuracy and training time. These tests were made after splitting 75% of the total number of the data in each corpus for training and 25% for testing data.

The results of DT classifier are better than NB classifier for OSN and reviews in general. The accuracies were higher and the F-measure was sometimes higher when the imbalance of data is extreme, otherwise it is convergent. NB is better for the benchmark balanced data. Using bigrams give better results than unigrams for benchmark data.

Applying text processing techniques increases the accuracy in some tests with NB classifier but it didn't affect much the DT classifier. The penalty in accuracy is not much but it gives the least training time.

The benchmark movie reviews corpus gives its highest accuracy and highest F-measure after replacing the negation words and the following negated words with the antonyms of the negated words when using NB and bigrams as features. The IMDB corpus gives its highest accuracy and highest F-measure after replacing the negation words and the following negated words with the antonyms of the negated words and removing stopwords when using NB and unigrams as features. Facebook corpus gives its highest accuracy when no text processing techniques is applied when using DT and unigrams as features while it gives its highest F-measure when using NB and unigrams with no text processing. The Twitter corpus gives its highest accuracy and highest F-measure with any combination of text processing and features when using DT classifier. This suggests that DT is better classifier for OSN unbalanced data while NB is better for reviews balanced data. The least training time was always achieved after applying all text processing techniques along with NB and unigrams for all corpora.

In the future we plan to repeat these tests on data with different languages than English e.g. Arabic language.

REFERENCES

[1] W. Medhat, A. Hassan and H. Korashy, "Sentiment analysis algorithms and applications: A survey," Ain Shams Engineering Journal, 2014.

[2] B. Liu, Sentiment Analysis and Opinion Mining: Morgan & Claypool Publishers, May 2012., 2012.

[3] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentimentclassification using machine learning techniques," in Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-2002), 2002.

[4] A. Kennedy, D. Inkpen, "Sentiment Classification of Movie Reviews Using Contextual Valence Shifters," Computational Intelligence, 2006.

[5] F.Benevenuto, G. Magno, T. Rodrigues, and V.Almeida, "Detecting Spammers on Twitter," presented at the CEAS 2010 Seventh annual Collaboration, Electronic messaging, AntiAbuse and Spam Conference, Redmond, Washington, US, 2010.

[6] Mullen, Tony and N.Collier, "Sentiment analysis using support vector machines with diverse information sources," in in Proceedings of EMNLP-2004, 2004.

[7] Xia, Rui and C. Zong, "Exploring the use of word relation features for sentiment classification," in in Proceedings of Coling 2010: Poster Volume, 2010.

[8] Xia, Rui and C. Zong, "A POS-based ensemble model for crossdomain sentiment classification," in in Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP-2010). 2011.

[9] M.A. Walker, P. Anand, R. Abbott, Jean E. Fox Tree, C.Martell, and J.King, "That is your evidence?: Classifying stance in online political debate," Decision Support Systems, vol. 53, pp. 719-729, 2012.

[10] W. Medhat, A. Hassan and H. Korashy, "A Framework of preparing corpora from Social Network sites for Sentiment Analysis " accepted in the International Conference on Information Society (i-Society 2014), London, UK, 2014.

[11] R. Mehta, D. Mehta, D. Chheda, C. Shah and P. M. Chawan, "Sentiment analysis and influence tracking using twitter," International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE), vol. 1, pp. -72, 2012.

[12] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in Proceedings of the Workshop on Languages in Social Media, 2011, pp. 30-38.

[13] X. Bai, R. Padman, & Airoldi, "Sentiment Extraction from Unstructured Text Using Tabu Search -Enhanced Markov Blanket," Technical Report CMU-ISRI-04-127, 2004.

[14] M. Gamon, "Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis," in Proceedings of COLING'04, Geneva, Switzerland, 2004, pp. 841-847.

[15] M. Wiegand, A. Balahur, B. Roth, D. Klakow, and A.Montoyo, "A Survey on the Role of Negation in Sentiment Analysis," in Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, Uppsala, 2010, pp. 60–68.

[16] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "WordNet: an on-line lexical database," Oxford Univ. Press., 1990.

[17] J. Perkins, Python Text Processing with NLTK 2.0 Cookbook. Birmingham: Packt Publishing Ltd., 2010.

[18] B. Santorini, "Part-of-speech tagging guidelines for the Penn Treebank Project," ed. University of Pennsylvania, School of Engineering and Applied Science, Dept. of Computer and Information Science., 1990.

[19] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Found. Trends Inf. Retr.,, 2008.

[20] F. Benamara, C. Cesarano, A. Picariello, D. Reforgiato, and V. S. Subrahmanian, "Sentiment analysis: Adjectives and adverbs are better than adjectives alone," in Proceedings of the International Conference on Weblogs and Social Media (ICWSM)'07, 2007.

[21] T. Nasukawa and J. Yi, "Sentiment analysis: Capturing favorability using natural language processing," in Proceedings of the Conference on Knowledge Capture (K-CAP), 2003.

[22] J. M. Wiebe, T. Wilson, R. Bruce, M. Bell, and M. Martin, "Learning subjective language," Computational Linguistics, vol. 30, pp. 277–308, 2004.

[23] .C. Aggarwal and C. Zhai, Mining Text Data. Springer New York Dordrecht Heidelberg London: © Springer Science+Business Media, LLC 2012, 2012, p.^pp. Pages.

[24] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," presented at the AAAI Workshop on Learning for Text Categorization, 1998.

[25] J. R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, pp. 81–106, 1986.

[26] Kotsiantis, B. Sotiris, I. D. Zaharakis, and P. E. Pintelas, "Supervised machine learning: A review of classification techniques," pp. 3-24, 2007.

[27] Chawla, V. Nitesh, N. Japkowicz, and A. Kotcz, "Editorial: special issue on learning from imbalanced data sets," CM Sigkdd Explorations Newsletter, vol. 6, pp. 1-6, 2004.

[28] David A. Cieslak, and N. V. Chawla, "Learning decision trees for unbalanced data.," Machine Learning and Knowledge Discovery in Databases, pp. 241-256, 2008.

[29] Ferri, César, P. Flach, and J. Hernández-Orallo, "Learning decision trees using the area under the ROC curve," ICML, vol. 2, 2002.

[30] N. V. Chawla, "C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure," in Proceedings of the ICML, 2003.

[31] S. C. Wei Liu, David A. Cieslak and N. V. Chawla, "Robust Decision Tree Algorithm for Imbalanced Data Sets," SDM, 2010, pp. 766-777.