Sentiment Analysis Methodology of Twitter Data with an application on Hajj season

Mahmoud Elgamal

The Custodian of the Two Holy Mosques Institute for Hajj and Omra Research, Umm Al -Qura University, Makkah, Saudi Arabia

Abstract**—** *With the rapid growth of the internet, millions of people are sharing their views and opinions on a variety of topics on microblogging sites, as it contains simple expressions. Microblogging websites are just social media sites to which user makes real time short and frequent posts about everything. In big event gathering like Hajj, to get rapid and accurate views and impressions of hajji about some quality of service or other views is of a great importance as time and space are limited. In this paper, we utilize tweets during Hajj to do sentiment analysis; the tweets are preprocessed by experience three phases; tokenization, normalization, and part of speech (POS) tagging. In the final step, Naïve Bayes classifier used to classify tweet as positive or negative by comparing each word in the query tweet with the labeled words in the lexicon.*

Keywords**— *Naïve Bayes Classifier, Performance Analysis, Sentiment Analysis, Twitter.***

I. INTRODUCTION

In the last few years, twitter has been hugely increased as a social network enables users to send and read 140 character messages in real time. Moreover, users can share their opinions about many topics, e.g., sports, social etc, discuss complains and express positive attitudes.

Inspired by the huge growth of twitter, companies and organizations are increasingly seeking ways to mine twitter for information about people's opinion about their services and products. In KSA, there are permanent Hajj and Umra seasons where people come to do their religious rituals; they stay more than weeks in KSA.

The Hajji express their impressions many aspects like hotels, transport.etc is an important source of information for decision maker if it is mined and analyzed to get the Hajji feedback about many topics.

Among uses of sentiment analysis, is that the business improvement of an organization can be tracked by user’s feedback [1, 3, 6].

This paper discusses twitter sentiment analysis in details for English language as it has a plenty of available resources. In the future work, it is planned to extend the work for Arabic language.

The paper is organized as follows: section 2 describe

system architecture, section 3 feature extraction of tweets data, section 4 classification, section 5 experiment, and conclusion in section 6.

II. SYSTEM ARCHITECTURE

Our system is organized in five main components: preprocessing of tweets, feature extraction, training set which is a set of predefined positive or negative tweets used for building the sentence database against it the classification of a query tweet is done, classifier using naive Bayes or support vector machines (SVMs), and the output(positive or negative).

These components are connected in pipeline architecture, see figure (1). The classifier determines the polarity class of the tweet message as a final output.

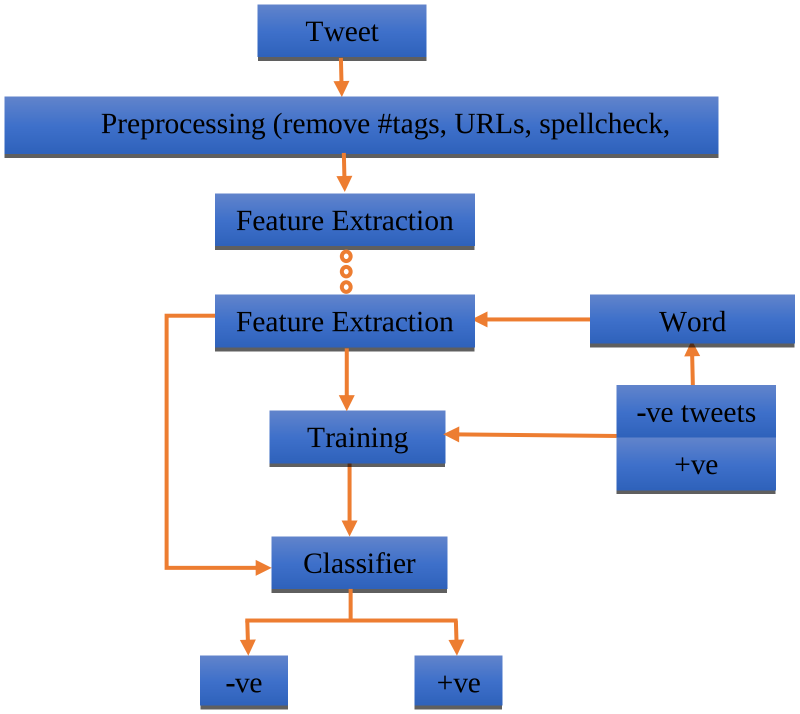


FIGURE (1) SYSTEM ARCHITECTURE

III. PREPROCESSING OF TWEETS DATA AND FEATURE EXTRACTION

It is well known that a tweet consists of 1) Emoticons: express the user's mood. 2) Target: symbol "@" used by twitter to refer to user. 3) Hashtags:"#" used to mark topics and increase tweet visibility. 4) Tweet: short message express user's opinion about some topic. Twitter data are prepared with the aid of two dictionaries, i.e., emoticon dictionary and acronym dictionary. The emoticon dictionary contains about 170 labeled emoticons [9]; ":)" labeled as positive and ":)" labeled as negative. The acronym dictionary [1] has 5148 acronyms, e.g., WOW(Wonder of Wonders).

Tweets handled as follows [1]:

1. Look up for emoticons and their sentiment polarity (positive, negative, or neutral) in emoticons dictionary.
2. Replace all URLs with a tag ||u||.
3. Replace targets(e.g. "@Mark") with tag ||T||.
4. Replace all negations by tag "Not".

After performing the former steps, we remove hashtags, URLs, and make spell check.

The next step is to make emoticons tagging and POS(part-of-speech) tagging, POS tagging is the most difficult part, as one have to assign it to each word in a sentence.

For example, a sentence like "Heat water in a large vessel" will be [heat(verb) water(noun) in(prep.) a(det.) large(adj.) vessel(noun)] words associated with their tags.

In the final step, POS used to build a sentence database, namely, verbs, adjectives, emoticons,..etc.

In information retrieval (IR), POS used to compute a term weights, which are mathematical computations of how informative words are, and constitute an integral part of the statistical modelling of documents by IR systems.

3.1 Construction of n-grams: set of n-grams can be made out of consecutive words. Negation words such as “no”, “not” is attached to a word which follows or precedes it. For example: “I do not like soda” has two bigrams: “I do+not”, “do+not like”, “not+like soda”. So the accuracy of the classification improves by such procedure, because negation plays an important role in sentiment analysis.

3.1.1 Example Bi-gram: let the sentence:

"Under committee rules, it went automatically to a subcommittee for one week."

We run a snippet of Python NLTK[7] code on it, to get the Bi-gram model with conditional frequency distribution (CFD) is shown in table (1).

TABLE 1
CFD OF THE SENTENCE BIGRAM MODEL

Bi-gram model	Conditional Frequency Distribution
('Under', 'committee')	0.0001
('committee', 'rules')	0.0001
('rules', ',')	0.0001
(',', 'it')	0.022727273
('it', 'went')	0.0001
('went', 'automatically')	0.0001
('automatically', 'to')	0.0001
('to', 'a')	0.0001
('a', 'subcommittee')	0.0001
('subcommittee', 'for')	0.0001
('for', 'one')	0.0001
('one', 'week')	0.0001
('week', '.')	0.0001

In order to know the probability of word x followed by word y in the corpus we need use bigram model.

IV. CLASSIFICATION

A sentiment classifier using the multinomial Naïve Bayes classifier was nominated for classification. Naïve Bayes classifier yielded better results than support vector machines [5].

Naive Bayes model is a simplest model for the categorization of the text this model works well. Naïve Bayes classifiers assume that the effect of a variable value on a given class is independent of the values of other variable. This assumption is called class conditional independence. Naïve Bayes classifier is based on Bayes' theorem and given by[8]:

$$P_{NB}(c|d) = \frac{(p(c)\sum_{i=1}^{m} p(f|c)p(f|c)^{ni(d)})}{p(d)}$$

where ***d*** denote the tweet, f represent a feature, ***ni(d)*** is the count of feature ***fi*** found in tweet ***d***, and ***m*** is the total number of features ***p(c)*** and ***p(f|c)*** are obtained through maximum estimates.

4.1 Example for classification: Assume that, we have the following sentences that will be used as a training set:

Training set = ['I like spring.', 'I do not like this cafe', 'I am tired of this stuff.', 'This is an amazing place!', 'I feel very good about these fruits.', 'This is my best work.', 'I can't deal with this', 'He is my sworn enemy!', 'My boss is horrible.', 'What an awesome view']

First, we label each sentence from training set as either positive or negative and arrange positive and negative sentences as follows:

training_data = [('I like spring.', 'pos'),

('This is an amazing place!', 'pos'),

('I feel very good about these fruits.', 'pos'),

('This is my best work.', 'pos'),

("What an awesome view", 'pos'),

('I do not like this cafe', 'neg'),

('I am tired of this stuff.', 'neg'),

("I can't deal with this", 'neg'),

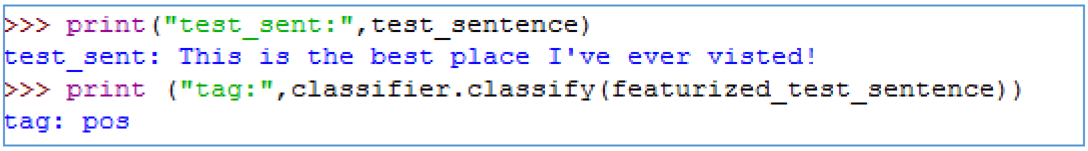
('He is my sworn enemy!', 'neg'),

('My boss is horrible.', 'neg')]

The classifier trained on the training_data and we tested it on the sentence:

test_sentence = "This is the best place I've ever visted!”

The output was:



```
>>> print("test_sent:",test_sentence)
test_sent: This is the best place I've ever visted!
>>> print ("tag:",classifier.classify(featurized_test_sentence))
tag: pos
```

FIGURE 2: CLASSIFICATION TEST

V. EXPERIMENT

The experiment done on a data[2] where the data consists of positive polarity of 5331 positive snippets and negative polarity of 5331 negative snippets.

Each line in these two files corresponds to a single snippet (usually containing roughly one single sentence); all snippets are down-cased, the snippets were labeled automatically. Then we run Naïve Bayes classifier on the data.

Next to evaluate the classifier performance, there are a number of other metrics used to evaluate classifiers; the most common are precision, recall and accuracy. To understand these metrics, we must first understand false positives (FP) and false negatives (FN). False positives happen when a classifier classifies a feature set with a label it should not have. False negatives happen when a classifier does not assign a label to a feature set that should have it. In a binary classifier, these errors happen at the same time.

We evaluate the performance of the classifier in terms of precision, recall, and accuracy [4] as:

- Precision (P) is the number of relevant documents retrieved by the system divided by the total number of documents retrieved (i.e., true positives plus false alarms).

$$P = \frac{TP}{TP + FP} \tag{1}$$

- Recall(R) is the number of relevant documents retrieved by the system divided by the total number of relevant documents in the database (which should have been retrieved).

$$R = \frac{TP}{TP + FN} \tag{2}$$

- Accuracy(A) is the probability that the retrieval is correctly performed

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where,

TP (True Positive) - correctly classified positive,

TN (True Negative) - correctly classified negative,

FP (False Positive) - incorrectly classified negative, and

FN (False Negative) - incorrectly classified positive.

The Naïve Bayes classifier run on the data to get the top (10,100, 1000, 10000, 15000) word features table (2) below and figures (2) & (3) depicts the result.

TABLE (2)
ACCURACY, PRECISION, AND RECALL OF TOP (10,100, 1000, 10000, 15000) WORD FEATURES.

# word features	Accuracy	precision (+ve)	recall (+ve)	precision (-ve)	recall (-ve)
10_WF	0.57464	0.54938	0.83046	0.65284	0.31883
100_WF	0.6823	0.65987	0.75244	0.71204	0.61215
1000_WF	0.79745	0.81695	0.76669	0.78021	0.82821
10000_WF	0.84696	0.86911	0.81695	0.82732	0.87697
15000_WF	0.84659	0.86378	0.82296	0.83095	0.87022

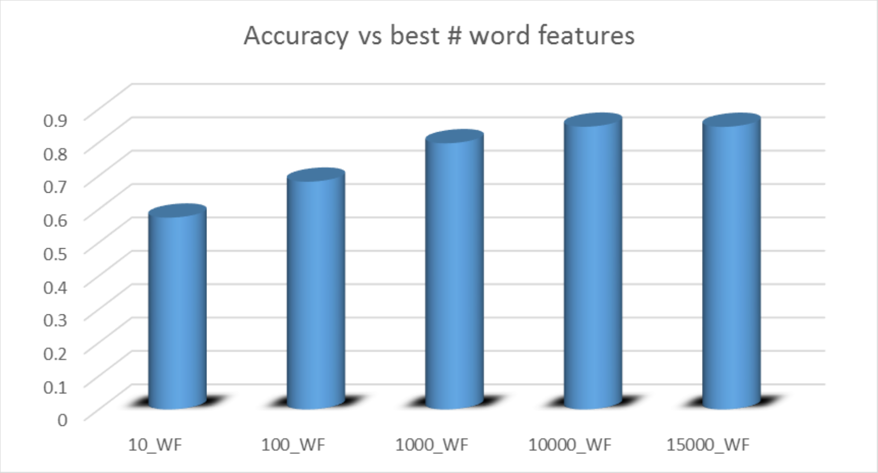


FIGURE (3): ACCURACY VERSUS NUMBER OF WORD FEATURES.

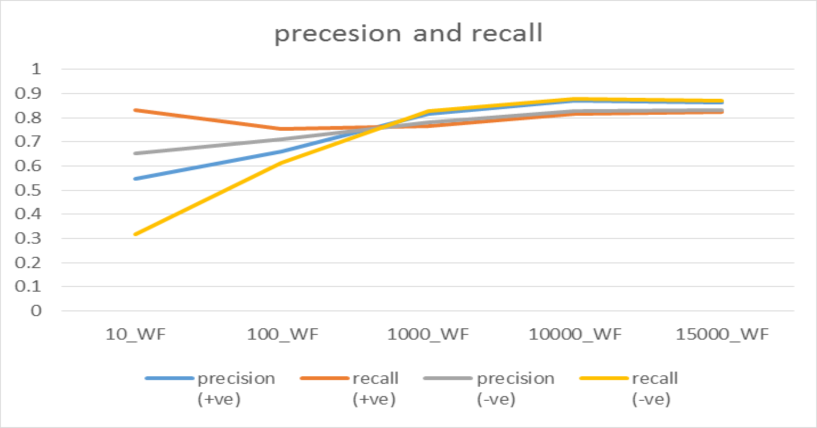


FIGURE (4): PRECISION AND RECALL OF POSITIVE AND NEGATIVE CLASSIFICATION.

VI. CONCLUSION

In this paper, we studied the methodology of sentiment analysis and the result was consistent for English corpus that was available for the study. We plan to do the same work for other languages in Hajj especially Arabic that are the majority of Hajji. Moreover, we plan to do our work on line in the next step.

REFERENCES

[1] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R., "Sentiment analysis of twitter data", In Proceedings of the Workshop on Language in Social Media (LSM 2011), pages 30–38, Portland, Oregon. Association for Computational Linguistics.

[2] AFINN Data Set 2011:
http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

[3] A. Kumar and T. M. Sebastian, "Sentiment Analysis on Twitter" department of Computer Engineering, Delhi Technological University, Delhi, India, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3, July 2012

[4] Olson, David L., and Delen, Dursun (2008), "Advanced Data Mining Techniques", Springer, 1st edition (February 1, 2008), page 138, ISBN 3-540-76916-1.

[5] Pak and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining", In Proceedings of the Seventh Conference on International Language Resources and Evaluation, 2010.

[6] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales ", Proceedings of the ACL, 2005.

[7] Natural Language Toolkit, http://www.nltk.org/

[8] V. Sahayak, V. Shete, and A. Pathan, "Sentiment Analysis on Twitter Data", IJIRAE, 2015.

[9] Wikipedia 2015, https://en.wikipedia.org/