Classification Analysis

1. Abstract

Twitter is a web application to determine online news and social networking service where users post and interact with messages, anywhere in the world. Twitter posts are generally short (140 characters in our data, 280 characters currently) and generated continuously by public which is well suited for opinion mining. Twitter messages can be classified either in positive or negative or neutral sentiment based on certain aspects with respect to term based query. The past studies of sentiment classification (Sentiment Analysis) are not very conclusive about which features and supervised classification algorithms are good for designing accurate and efficient sentiment classification system. We propose to combine many feature extraction techniques design a more accurate sentiment classification system.

2. Introduction

Sentiment Analysis, as Wikipedia states, 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 to be covered by reviews”. Sentiment Analysis identifies the sentiment expressed in a text then analyzes it. Therefore, the target of Sentiment Analysis is to find opinions, identify the sentiments they express, and then classify their polarity.

The data sets used in Sentiment Analysis are an important issue in this field. The main sources of data are from the product reviews. These reviews are important to the business holders as they can take business decisions according to the analysis results of users’ opinions about their products. The reviews sources are mainly review sites. Sentiment Analysis is not only applied on product reviews but can also be applied on stock markets, news articles, or political debates. In political debates for example, we could figure out people’s opinions on a certain election candidates or political parties. The election results can also be predicted from political posts. The social network sites and micro-blogging sites are considered a very good source of information because people share and discuss their opinions about a certain topic freely.

Twitter is a social networking web application with microblogging feature that has a large and constantly growing user data-base. Thus, the application provides a data set in the form of messages that are usually short status updates from Twitter application users. On Twitter, data that consists of millions of short messages and user status updates are generated each day on about hundreds of different topics. The task of extracting data from these small texts has become immensely useful for sorting and ranking popularity of topics mentioned within the updates. Nowadays twitter has emerged as one of the most popular platforms for expressing sentiments and thoughts on Internet. It is very useful and obvious to mine and analyses Twitter data for interesting information regarding major trending topics in the media and other spaces.

3. Methodology

Sentiment Analysis techniques can be roughly divided into machine learning approach, lexicon based approach and hybrid approach. The Machine Learning Approach (ML) applies the famous ML algorithms and uses linguistic features. Here, I have used three famous Machine Learning models which are:

- Multi-dimensional Naïve Bayes.
- Logistic Regression.
- Support Victor Classifier (SVC).

Each one of these models is generally divided into two/three categories. Each category represents the way the features are selected:

- Using only the dictionary of the training data (Dictionary-based Approach).
- Using the dictionary and the lexicon of the training data (Lexicon-based Approach).
- Using word2vec, the dictionary and the lexicon of the training data (Hybrid-based Approach).

The Lexicon-based Approach relies on a sentiment lexicon, a collection of known and precompiled sentiment terms. It is divided into dictionary-based approach and corpus-based approach which use statistical or semantic methods to find sentiment polarity. The hybrid Approach combines both approaches and is very common with sentiment lexicons playing a key role in the majority of methods.

The lexicon-based approach depends on finding the opinion lexicon which is used to analyze the text. There are two methods in this approach. The dictionary-based approach which depends on finding opinion seed words, and then searches the dictionary of their synonyms and antonyms. The corpus-based approach begins with a seed list of opinion words, and then finds other opinion words in a large corpus to help in finding opinion words with context specific orientations. This could be done by using statistical or semantic methods.

3.1. Tweets Preprocessing

Twitter is an online social networking service where users post and interact with messages. Users access Twitter through its website interface, Short Message Service (SMS) or mobile device application software. It's made so the people can say whatever they want in any way possible. Most of the people use informal ways to express what's in their mind which can make our models suffer when learning. We will try to take a few steps towards formal tweets by performing the following processes:

- Convert the internet slangs to normal words. So, the expression "12be" will be changes to "I want to be", the expression "4u" will be changes into "for you" and so on. We are an online corpus.
- Replace the emoticon icons with its polarity. So, the emoticon ":)" will be changed into the word "positive" and the emoticon ":(" will be changed into the word "negative" and so on.
- Formalize the verbs that has apostrophe negation. Like the verb "haven't" will be change to "have not", and "hasnt" will be changed to "has not". We are going to that in the following step.
- Handle the negation as stated by the Yahoo team which is to (add NOT_ to every word between negation and the following punctuation". So, the phrase "I didn't like this movie. The movie was …" will turn into "I didn't NOT_like NOT_this NOT_movie The movie was…".

- Lemmatize the tweets, so the verb “flies” will be “fly” and the word “networks” will be “network”.
- Remove punctuations, numbers, tags, websites url, emails, hashtags.
- Remove stop words using provided by the NLTK library.

3.2. Lexicon Features

After preprocessing tweets, now it’s time to extract the features. Extracting lexicon features is done via two consecutive processes. The first process is a dictionary-based approach, we use the Count Vector of and TFIDF vector of the training tweets. These vectors are formed using three language models combined (unigram, bigram, and trigram). The Count Vector doesn’t affect that much in the performance, so I have used SVD Truncated model to dimensionally-reduce the size which can reduce time and computational complexity.

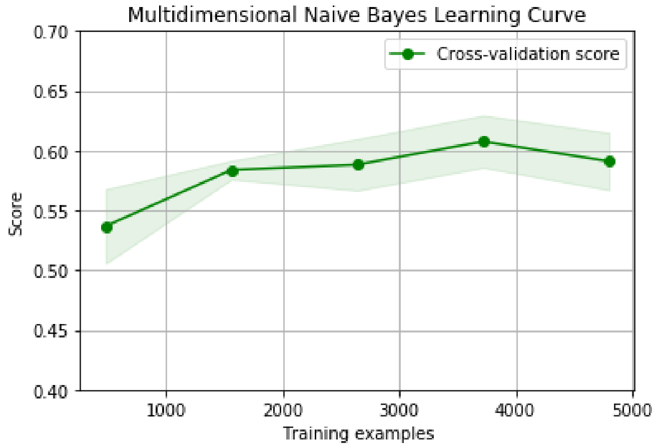
The second process is using the mutual information measure provides a formal way to model the mutual information between the features and the classes. This measure was derived from the information theory. The point-wise mutual information (PMI) between the word w and the class i is defined on the basis of the level of co-occurrence between the class i and word w. The expected co-occurrence of class i and word w, on the basis of mutual independence, is given by $P_i \,.\, F(w)$, and the true co-occurrence is given by $F(w) \,.\, p_i(w)$. I have used multiple corpora for scoring our tweets based on the PMI value. These corpora are:

- MPQA: corpus that contains the polarity of a huge amount of English words.
- Bing Liu: corpus about 6800 English words that are divided as either positive or negative.
- Afinn: The AFINN lexicon is a list of English terms manually rated for valence with an integer between -5 (negative) and +5 (positive) by Finn Årup Nielsen.
- SemEval2015: the corpus that the winning team NRC-Canada has used to score tweets in the SemEval2015 competition.
- WordSat: corpus has polarity English lexicon.
- SentiwordNet: corpus which has words with their polarity score.

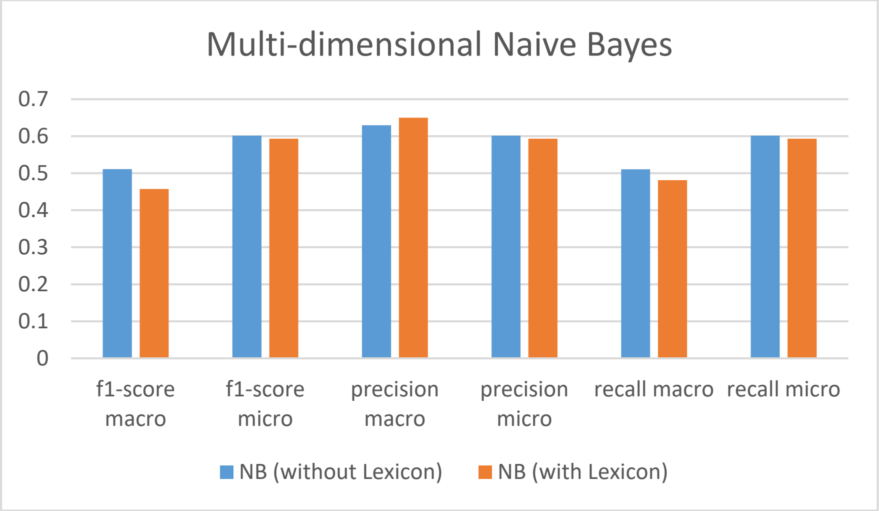
I have used the scores beside the number of mentioned words for each corpus. So, for each corpus of the six previous corpora, I have counted how many times a word has been substituted by a value inside the corpus and provided that as a feature.

4. Models

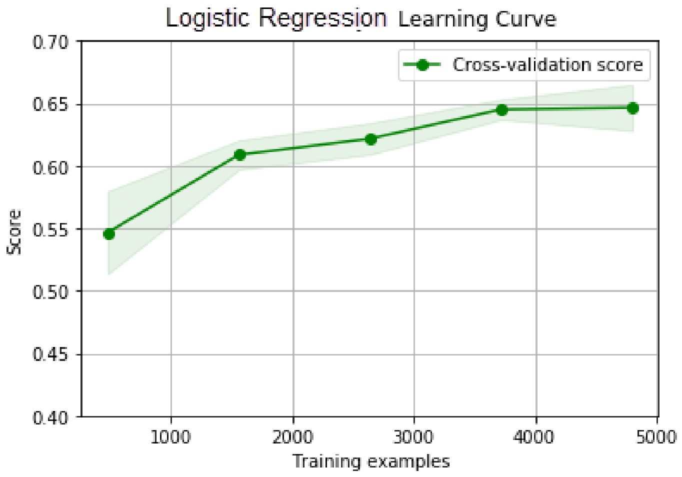
As I stated before, I have used three different models. The first model is the Multidimensional Naïve Bayes. This algorithm is a probabilistic classifier in a simple form that counts the combinations of values and frequency in a data set under consideration and calculates probabilities set. Bayes theorem is the base of this algorithm and assumes that all the attributes are completely independent against a set value of the class variable. I have made two Multidimensional Naïve Bayes models, the first without using any polarity lexicon and the second with polarity lexicon. The following figure is showing the learning curve for the two Multidimensional Naïve Bayes models:



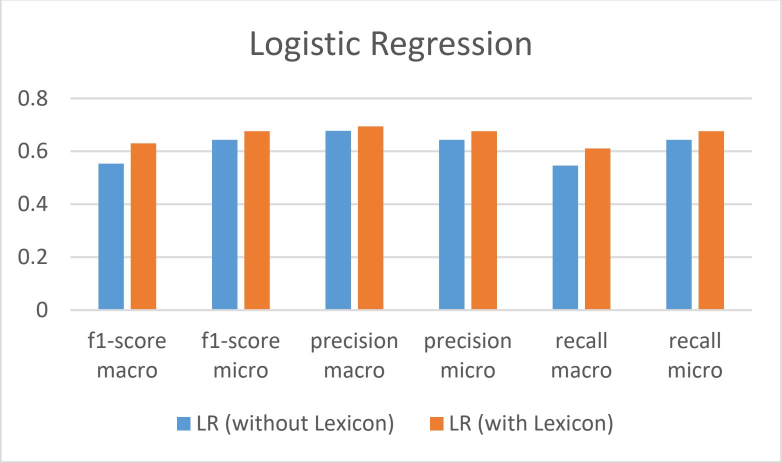
And this is different score metrics of the same model using different set of features:



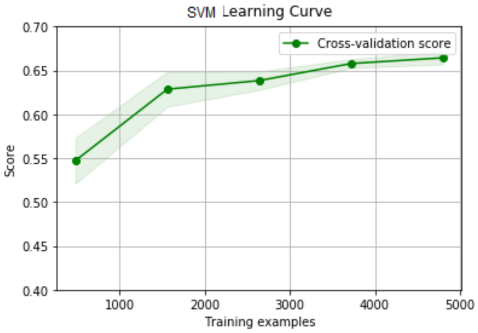
The second model I have used is the Logistic Regression model which is also has been used with two different sets of features. The first set using only the dictionary-based features, and the second set using all stated features. The following is the learning curve for our two models:



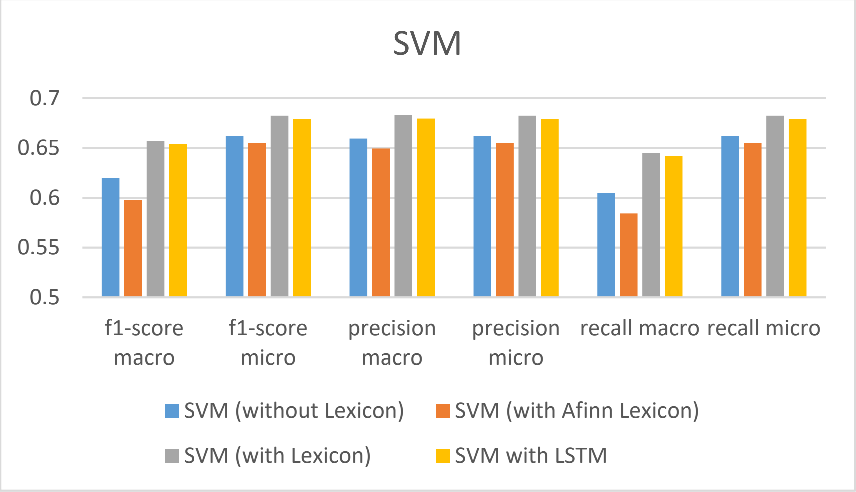
And this is different score metrics of the same model using different set of features:



The last model is "Support Vector Machine Classifier" which is the best model accuracy-wise. So, I have used it with five sets of features. The first set using only the dictionary-based features, the second set using the dictionary-based features beside Afinn polarity feature with the count. The third set using all lexicon features. The fourth set using all lexicon features plus features from one-layer neural network consisting of LSTM cells initialized with "GloVe50B50d" word vectors. And the last set is exactly like the previous one but with a three-layer neural network.

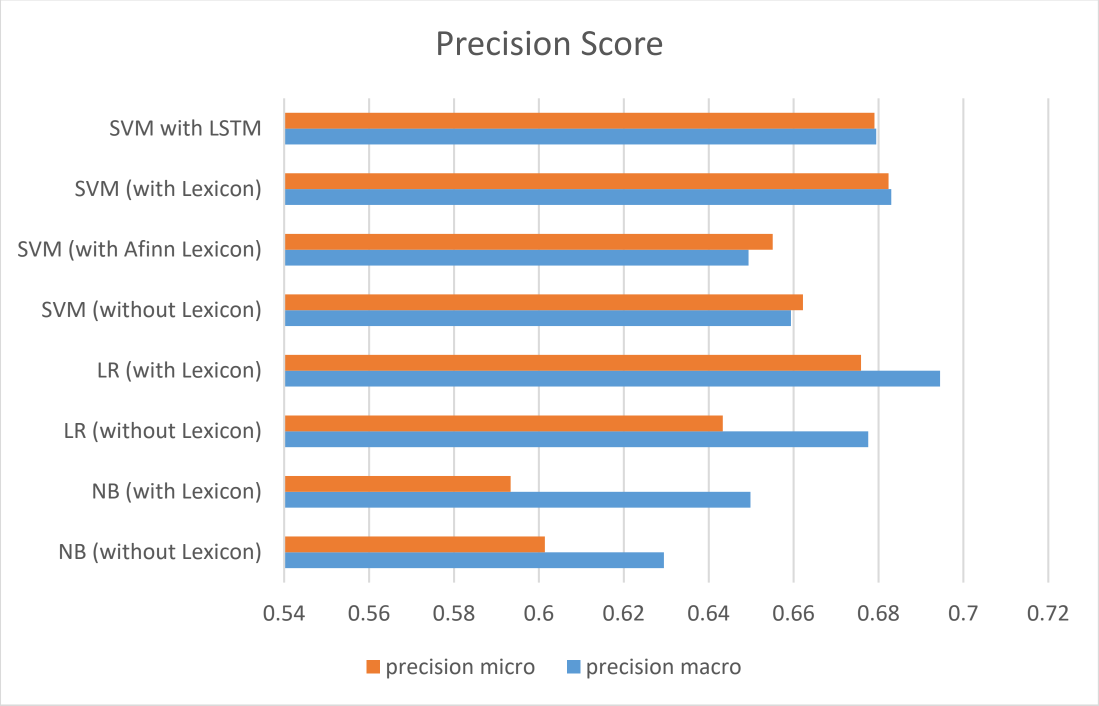


And this is different score metrics of the same model using different set of features:

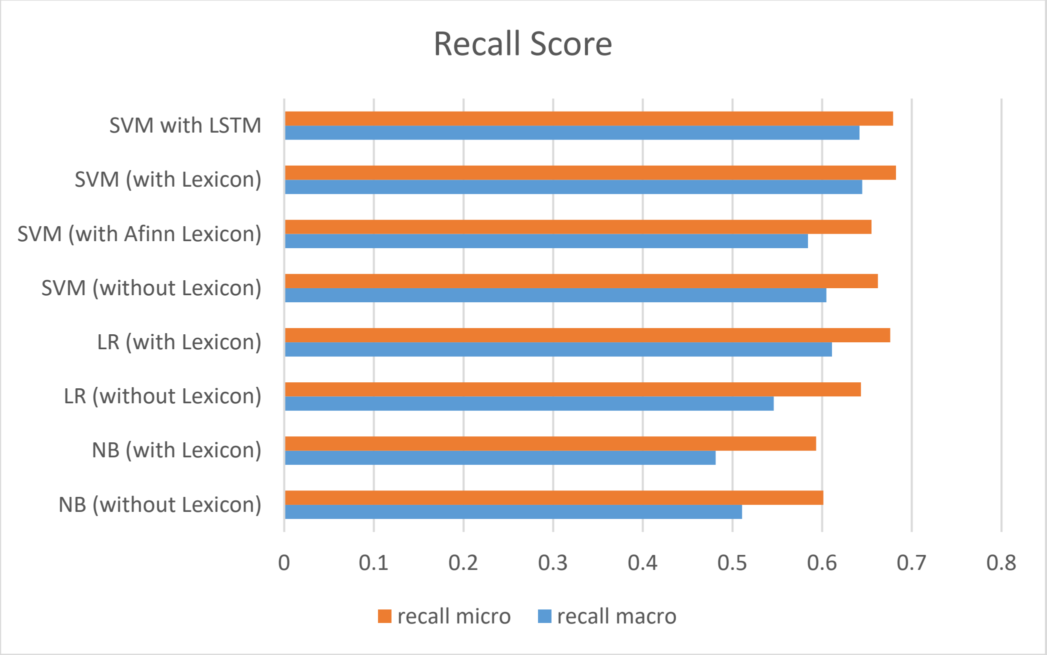


The following is a comparison between the all models according to:

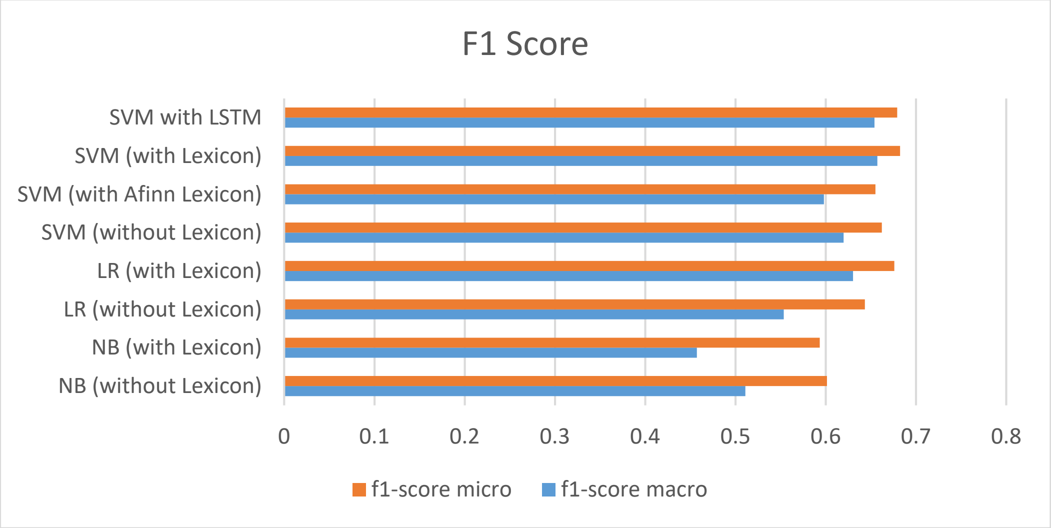
- Precision:



- Recall:



- F1-measure:


F1 Score
SVM with LSTM
SVM (with Lexicon)
SVM (with Afinn Lexicon)
SVM (without Lexicon)
LR (with Lexicon)
LR (without Lexicon)
NB (with Lexicon)
NB (without Lexicon)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
f1-score micro
f1-score macro