A Study using Support Vector Machines to Classify the Sentiments of Tweets

Wassim A. Zgheib
Department of Computer Science
American University of Science and Technology
Beirut, Lebanon

Aziz M. Barbar
Department of Computer Science
American University of Science and Technology
Beirut, Lebanon

ABSTRACT
It is difficult to sidestep Big Data today, as the industry is abuzz with its promises. The trend is towards data-driven decision-making in all aspects of businesses because making sense out of data is very profitable and valuable. People tend to use social media, especially Twitter, to tweet about their opinions and sentiments. However, due to the prevalence of data that might be noisy, varied, unfiltered, and the impractical state of manually labeling large number of tweets to train classifiers, data acquisition for training sentiment analysis classifiers is becoming more and more of a challenge. This paper proposes a solution to easily acquire automatically labeled, filtered, and huge training data from Twitter in order to be given as input to a support vector machine classifier. The recommended solution discusses the workaround of unlabeled data through using Twitter hashtags to automatically induct the sentiment of a tweet (positive or negative). Neutral class is trained using tweets generated by newspapers accounts. A test study was conducted to show the accuracy of the applied features on the classifier. As a result, tweets trending on Twitter can now be analyzed to induce their sentiments which helps organizations in future data-driven decisions.

Keywords
Big Data, data-driven, Twitter, automatically labeled, training, support vector machine, unlabeled data, hashtags, newspapers.

1. INTRODUCTION
The impact of technology has exploded data in the world. Big Data is the term used to refer to the explosive amount of data generated that is beyond the current storage and processing capacity. Learning from this data is becoming crucial where data-driven decisions makings are a necessity in all aspects of businesses [1]. Many organizations are transforming their business using machine learning techniques because making sense out of the data is very profitable, valuable, and whatever the business type is, opinion mining of products is crucial to the business [1].

Twitter is part of Big Data as 500 million tweets are generated every day [2]. Twitter is a very popular social media website where users can create accounts and post short messages – limited to 140 characters – called tweets. Followers of each account can see and retweet the tweets posted by this account. People tend to use Twitter to tweet about their sentiments and opinions related to products, persons, and events. As stated, learning from these tweets is very crucial to any type of business. There are a lot of studies that focus on text sentiment analysis. In addition, there are data sets that are already built and ready to be used. However, sentiment analysis on social media portals is a recent topic. Many challenges are faced in the process of knowledge discovery from Twitter [3]. Messages length limitation in Twitter, grammatical mistakes, unfiltered tweets, noisy tweets, and the impractical state of manually labeling tweets to train classifiers constitute the main challenges in performing sentiment analysis on Twitter [3].

This paper proposes an application that easily acquires automatically labeled, filtered, and huge training data from Twitter in order to be given as input to a support vector machine classifier. The application has two phases, one for training the classifier where tweets are gathered from Twitter API, and one for predicting the classes of new tweets as positive, negative, or neutral. The study demonstrates the efficiency of using distant supervision method to induct the sentiments of tweets through using specific hashtags to train the positive and negative classes. Neutral classes were trained through using Twitter newspapers accounts.

After performing analysis on tweets features, 1-grams and part-of-speech tags were applied as features to be implemented on a support vector machine classifier. In the end, the efficiency of the results is demonstrated by conducting a test study that shows the accuracy applied on a randomly formed test data set of tweets. This test data set is manually labeled.

The coming section discusses the related work. Then, the proposed solution is stated in terms of data gathering phase, and classification phase. Details of the implemented Support Vector Machine classifier are explained along with libsvm.net library used and the features analysis phase. After that, a test study of manually labeled tweets will be conducted to show the accuracy of the results. In the end, the work of this study will be concluded along with future work to enhance the results obtained.

2. RELATED WORK
In this section, a brief overview of some existing studies is presented. These studies used distant supervision concept on Twitter to induct the sentiments of tweets. These related studies differ in their accuracy achieved, the algorithms used, the number of tweets used to train the classifier, and the methods applied to automatically classify training data.

2.1 Twitter as a Corpus for Sentiment Analysis and Opinion Mining
A corpus of 300,000 text posts is collected that is split in an even way among three classes: text containing positive emotions, text containing negative emotions, and neutral texts (objective texts). Sentiment Classifier will be trained with posts of the corpus collected. These 300,000 text posts used as training data should be classified first. However, it is time consuming to manually label these data. So, in this study [4], a method is proposed to automatically label this data without human intervention. The method suggests using emoticons in order to automatically classify text posts as positive or negative. Most of the tweets are composed of a single

sentence, and then an emoticon within this sentence represents the sentiment of the whole sentence. Using Twitter API, a set of text is collected that contain two types of emoticons:

Positive Emoticons: $:-),:),:D,=),$ etc.

Negative Emoticons: $:-(,:(,=(,;(,$ etc.

In order to obtain neutral texts posts, to train classifier on objective texts, newspapers' Twitter accounts are chosen as a source for training data, like: "New York Times", "Washington Posts" etc. 44 newspapers' accounts were queried. English language is used in this study.

"TreeTager" is used to tag all posts in the corpus. Features of positive, negative, and neutral sets can be analyzed and extracted from part of speech tags results in order to train the sentiment classifier. N-gram (sequence of n words) is used as a binary feature. Unigrams, bigrams, and trigrams were used, and it's notable that trigrams should better capture patters of sentiments expressions [4]. Unigrams provide good coverage of the data. There is a filtering phase where some texts are removed: URL links, Twitter user names, Twitter special words such as "RT", and emoticons. Text Segmentation – tokenization – is done where a space or punctuation marks are found. Short words like "don't", "I'll" are not tokenized, and they remained one word. Stopwords like "a", "an", "the" are also removed. The sentiment classifier is built using multinomial Naïve Bayes classifier. Naïve Bayes classifier relies on the calculation of conditional probability of the class giving the attributes. This is equivalent to the calculation of the probability of the set giving the twitter message.

After performing features analysis of part of speech tags, it is remarkable that best tags that can be used as features to classify objectivity from subjectivity are: adjectives, personal pronouns, and proper nouns. Whereas comparing positive class to negative class, the best part of speech tags features were: adverbs, verbs in past [4].

The classifier has been tested on 216 tweets manually labeled (108 positives, 75 negatives, 33 neutrals). Results show higher accuracy when using bigrams, when adding the attachment of negation words, and when discriminating common datagrams [4].

2.2 Twitter Sentiment Classification using Distant Supervision

This classification phase uses Twitter as a corpus and uses the method of automatically labeling data – distant supervision – using emoticons [5]. Neutral sentiments are not taken into consideration and thus two classes are used: positive, negative. Previous study [4] was fetching data in a random way. This study [5] queries data based on a query term. API calls are sent periodically to retrieve tweets with positive emoticons and negative emoticons used to train the classifier. The main approach in this study [5] is using four machine learning classifiers: Keyword-based, Maximum Entropy, Naive Bayes, and Support Vector machines. The study does a feature reduction approach. It replaces all mentioned usernames with "USERNAME" word, and all URLs with "URL" word. In addition, any letter occurring more than two times in a row is replaced with two occurrences. After performing these features reductions, the feature set down to 45.85% of its original size. Tweets containing both positive and negative emoticons are removed in order to obtain accurate results in the training phase, and retweeting tweets are removed also. After processing the data, 800,000 positive tweets and 800,000 negative tweets were fetched, resulting in total of 1,600,000 training tweets [5].

Using Twitter API, test data is retrieved using queries of various domains (products, companies, movies, people. locations…). 177 positive tweets and 182 negative tweets are manually retrieved resulting in total of 359 test tweets. Results showed that SVM had higher accuracy (82.2 %) when trained only with unigrams. The second higher accuracy achieved (81.9 %) was when SVM was trained with unigrams + part of speech tags [5].

2.3 Distant Supervision for Tweet Classification Using YouTube Labels

This recent study [6] suggested a novel approach for automatically labeling training data. The idea was to retrieve tweets containing URLs linking to YouTube videos. Videos on YouTube are already manually labeled in terms of categorizations (18 categories), for instance, education, entertainment, etc. and thus, each tweet is known automatically to which category (class) it belongs, and can be used as a training data. Accuracy obtained was 61.1%.

2.4 Enhanced Sentiment Learning Using Twitter Hashtags and Smileys

This study [7] relied on 50 Twitter hash tags and 15 smileys as automatic sentiment labeling. For instance, #sad for sure include all negative tweets. #happy includes all positive tweets, and so on. But neutral class was not taken into consideration in this study. Features used in this study were: punctuation, words, n-grams and patterns. Accuracy in this study is in terms of how well the use of hashtags and smileys can distinguish between sentiment types. On average, the percentage was greater than 80%.

This section presented some of the related studies that used distant supervision concept to induce the sentiments of tweets. Existing studies varied in the use of the classifier's algorithm, methods to automatically label data, size of training data set, and accuracy results. Next section summarizes data gathered used for training the classifier.

3. DATA GATHERING PHASE

This phase of the application easily acquires large amount of data from Twitter after implementing all required details documented in Twitter API. Due to the time consuming manner of manually labeling data, the application enables automatic classification of training data through specifying specific Twitter hashtags.

Latest tweets retrieved related to a specific hashtag or Twitter account can be saved for training, or for testing. This proposed application enables visual interaction with the data before being saved. The user decides what hashtag/username to use, the count of tweets to retrieve, and the class (positive, negative, neutral) of data retrieved before being saved into a file. The application performs a huge amount of data filtering before being saved (more about this phase in Data Filtering section).

Training data is gathered through specific adjectives that represent positive and negative polarities. These adjectives were taken from the multi-perspective question answering (MPQA) Opinion Corpus that has a list of positive and negative adjectives [8]. To prove the efficiency of using distant supervision concept in classifying data, specific Twitter hashtags and accounts were used to retrieve automatically labeled tweets. These keywords are listed in

table 1. Total of 3000 tweets were gathered from each of the three classes building a data set composed of 9000 automatically labeled tweets that will be used to train the classifier.

Table 1. Hashtags and Accounts Used for Training

Positive Hashtags	Negative hashtags	Neutral Accounts
amazing	bad	Mashable
splendid	worse	cnnbrk
good	worst	Big_Picture
better	awful	theonion
best	ugly	time
wonderful	wasteful	breakingnews
fantastic	difficult	bbcbreaking
comfortable	uncomfortable	espn
valuable	useless	harvardbiz
worthy	disgusting	gizmodo
useful	pointless	techcrunch
love	hate	wired
important	dislike	wsj
lovely	angry	nytimes
beautiful	baffled	foxnews
brilliant	confused	
essential	banal	
excellent	beastly	
Ilike	ihate	
great	notgood	
astonishing	bogus	
like	dontlike	
acceptable	boring	
admirable	annoying	
adorable	irritating	
affordable	obnoxious	
appropriate	hideous	
beneficial	invalid	
betterthanexpected	tedious	
likeit	waste	
simple	complex	
awesome	lost	

3.1 Data Filtering

Data have to be filtered and preprocessed in a suitable format for training. Data filtering is applied in a way to remove everything that do not help classifier in making classification. Reducing the training data set help in achieving higher accuracies. Following are the filters applied on all the tweets retrieved:

1) Remove empty spaces from beginning, middle, and end of a tweet.
2) Remove any special character that is not an English letter or a number. (Numbers are important due to their heavy use in neutral tweets when they're informative).
3) Converting tweet text to lower case.
4) Remove retweet symbols.
5) Remove URLs symbols and links.
6) Replace more than 3 times repeated characters with only 2 occurrences.
7) Remove any word that does not help in analyzing the sentiment of a tweet. These words are called stopwords [9].
8) Remove duplicate tweets.

3.2 Classification Phase

After data have been preprocessed and inserted into a .csv file for training, this file is given as input to a support vector machine classifier implemented in C#. The application generates corresponding vocabulary and builds a classification problem using 1-grams and part of speech tags as features for texts (more about this in section 4 – Features Analysis)

The classification phase transforms the training data in the .csv file into a format suitable to be read by "libsvm.net" library [10] that implements support vector machine algorithm. This library is implemented in National Taiwan University (more about this library in section 4 – Libsvm.net).

This section explained the details for gathering automatically labeled tweets through using the list of hashtags in Table 1. Filters done on tweets were also stated. The next section will handle the implementation details of the classifier.

4. IMPLEMENTATION

This section starts by explaining in brief Support Vector Machines (SVM) algorithm and classification type. Then, "libsvm.net" library classification type and data format are stated. In the end, features are analyzed in order to be used in classification.

4.1 Support Vector Machines

Support Vector Machines algorithm tries to find the optimal hyperplane that maximizes the margin of training data points. When data are not linearly separable, data instances are transformed into higher n-dimensional features space where they become linearly separable, using the kernel techniques [11].

Support Vector Machines are basically used in binary classification. However, they can be extended to multiclass classification problems using the one-against-one approach. This approach [12] states to build one SVM classifier for each pair of classes through training $K\,(K-1)\,/\,2$ classifiers where k is the number of classes. Then, each classifier trains data from two classes. This means that each classifier performs binary classification of the classes. At prediction time, a voting scheme is applied: all classifiers are applied to a new sample and the class that got the highest number of predictions gets predicted by the combined classifier. The concept of this approach is based on highest probability between classifiers, in contrast to "one-against-all" approach, which consists of building one SVM classifier per class. And then, this classifier learn through distinguishing the samples in its class from the samples in all remaining classes [12].

In this study, SVM is used in the purpose of multiclass classification. It uses the one-against-one approach that is implemented in libsvm.net library. The implementation of SVM is downloaded from an online tutorial written by Alexandre Kowalczyk [13].

Support Vector Machines algorithm was chosen because it is the best algorithm when it comes to text classification [14]. The reason is that texts produce high features space (more than 10000 features), and Support Vector Machines are the best in such cases as data are transformed easily into n-dimensional feature space using the kernel function in order to be linearly separated [14].

4.2 Libsvm.net

"libsvm.net" supports various Support Vector Machines formulations for classification, regression, and distribution [10]. This study is based on classification that has many types in the library. C-Support Vector Classification (C-SVC) was used as it supports multi-classification and is flexible by changing the value of the regularization parameter C. This parameter's value specifies the amount of misclassification data to be avoided. Sometimes, a smaller hyperplane can do a better job in correctly classifying data, and this corresponds to high value of C. In the used classifier, C was set to 100.

"libsvm.net" format means that your document needs to be pre-processed [10]. One training record is a list of nodes where each node has its own ID (index) and its value, as follows:

$$< index1 >:< value1 >< index2 >:< value2 > \cdots < indexN >:< valueN >$$

where N is the number of features.

4.2 Features Analysis

In order to obtain high classification accuracy, sentiment features of tweets must be analyzed to best separate the classes. The applied features are unigrams and the part-of-speech (POS) tags which showed efficiency when applied as features [15]. The algorithm was implemented using "OpenNLP" Library [16] that exports each part-of-speech tag for all texts words gathered using the hashtags and accounts in Table 2. A fair number of 200 tweets per hashtag and account were taken into consideration in order to analyze differences between subjective and objective tweets.

Table 2. Hashtags & Accounts Used in Features Analysis

Positive Hashtags	Negative Hashtags	Neutral Accounts
happy	sad	Mashable
sohappy	sosad	cnnbrk
glad	depressed	Big_Picture
motivation	stress	theonion
smile	angry	time
positivethoughts	hopeless	breakingnews
thinkpositive	Anxious	bbcbreaking
cheerful	disappointed	espn
amazing	bad	harvardbiz

Results in "Figure 1", "Figure 2", and "Figure 3" show that positive and negative classes have higher occurrences of "JJ" tag, while neutral class has higher occurrences of "NN" & and "NNS" tags where:

- "JJ": Adjective
- "NN": Noun, singular or mass
- "NNS": Noun, plural

Results shows that "JJ" tag occurred 2370 times and 2339 times in the positive and negative classes respectively, while it occurred 1669 times in the neutral class.

On the other hand, "NN" tag occurred 4871 times and 4879 times in the positive and negative classes respectively, while it occurred 5868 times in the neutral class. Same for "NNS" tag, it occurred more in the neutral class.

Thus, the occurrence of these 3 tags are used as features in the Support Vector Machine classifier in order to distinguish positive or negative classes from the neutral class. 1-grams are also the features mainly applied to distinguish positive classes from negative classes.

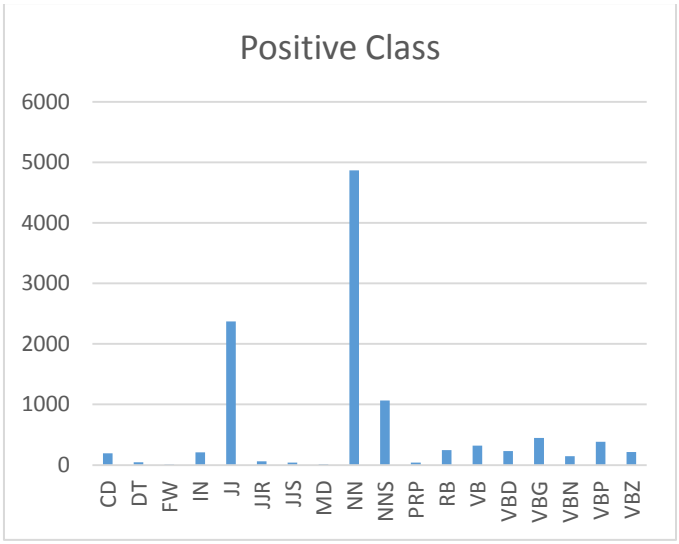

Fig 1: POS tags results in positive class

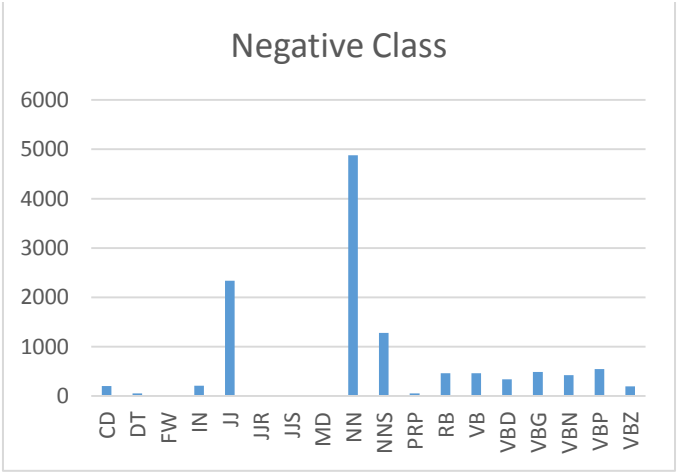

Fig 2: POS tags results in negative class

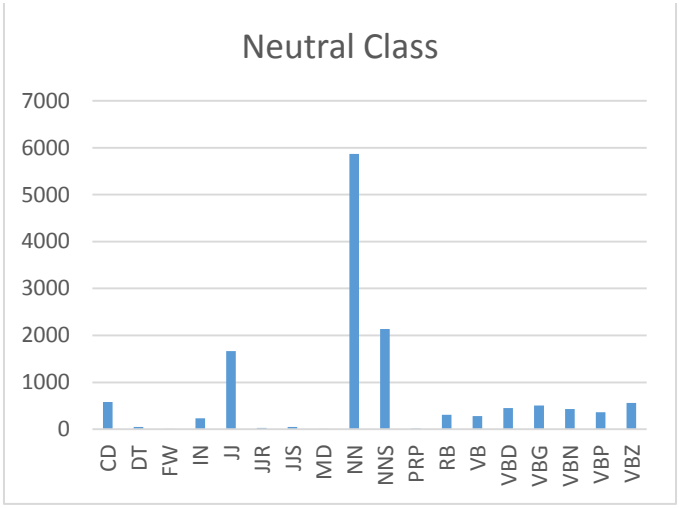

Fig 3: POS tags results in neutral class

5. RESULTS

The testing phase consists of manually labeling a set of tweets to build a test dataset. In order to obtain a testing dataset, a list of texts should be manually classified as positive, negative, or neutral, and then, they should be given as input to the trained classifier. After the classifier has been trained with 9000 records (tweets) obtained directly from Twitter API, testing is loaded.

A random list of 100 manually classified records has been applied to the trained model. No specific strategy was applied to the testing records, and the number of records from each class is also random. Out of the 100 manually labeled records, 85 records have been correctly classified by the classifier. Only 15 records were misclassified which corresponds to an accuracy of 85%.

Some of the misclassified records are inevitable, as some neutral records may contain also adjectives. For instance: "he has made an important decision". The adjective important can be used to express positive feedbacks and can be used in neutral scenarios also. Other issue can be when the sentence is neutral, but there is a specific sentiment behind it, for example: "#windows10 has bugs".

6. CONCLUSIONS AND FUTURE WORK

The paper's work highlighted the efficiency of using automatically labeled data in Machine Learning to classify new unlabeled data. Multi-class classification is in terms of sentiment analysis, where the classes are: Positive, Negative, and Neutral. The efficiency is in terms of time and cost as the proposed application easily gathers and filters huge amount of automatically classified tweets. The novel work in this application is ability to easily build a huge and ready data set of tweets in a small amount of time. Twitter was used as a corpus for training the classifier as it is the most used micro blogging website in the world.

The accuracy achieved was 85% on a test data set. What is worth to mention is that users of this application can gather tweets also of any Twitter hashtag or account, save them in a file, loads the file into the Sentiment Analysis application, predict data, and export the results to Microsoft Excel in order for concerned people to analyze the feedback on Twitter.

As a future work, the obtained results should be improved through analyzing more the features of tweets and adding more features (2 & 3-grams) without affecting the performance of the training phase. Stemming also will be added on all words to enhance performance [17].

7. REFERENCES

[1] S. F. Wamba, S. Akter, A. Edwards, G. Chopin and D. Gnanzou, "How 'big data' can make big impact: Findings from a systematic review and a longitudinal case study," International Journal of Production Economics, vol. 165, pp. 234-246, 2015.

[2] Kit_Smith, "Marketing: 96 Amazing Social Media Statistics and Facts for 2016," Brandwatch, 7 March 2016. [Online].

[3] S. Rosenthal, P. Nakov, S. Kiritchenko, S. M. Mohammad, A. Ritter and V. Stoyanov, "SemEval-2015 Task 10: Sentiment Analysis in Twitter," in Proceedings of the 9th International Workshop on Semantic Evaluation, Denver, 2015.

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

[5] A. Go, R. Bhayani and L. Huang, "Twitter Sentiment Classification using Distant Supervision," Stanford, 2009.

[6] W. Magdy, H. Sajjad, T. El-Ganainy and F. Sebastiani, "Distant Supervision for Tweet Classification Using YouTube Labels," in Ninth International AAAI Conference on Web and Social Media, Oxford, 2015.

[7] D. Davidov, O. Tsur and A. Rappoport, "Enhanced Sentiment Learning Using Twitter Hashtags and Smileys," in Proceedings of the 23rd international conference on computational linguistics, 2010.

[8] T. Wilson, J. Wiebe and P. Hoffmann, "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis," in Proceedings of the conference on human language technology and empirical methods in natural language processing, 2005.

[9] S. Allen, "stopword-dictionary," 2007-2016. [Online].

[10] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, 2011.

[11] D. Meyer, Support Vector Machines ∗ The Interface to libsvm in package e1071, Technikum Wien, 2015.

[12] J. Milgram, M. Cheriet and R. Sabourin, "One Against One" or "One Against All": Which One is Better for Handwriting Recognition with SVMs?," in enth International Workshop on Frontiers in Handwriting Recognition, 2006.

[13] A. KOWALCZYK, "How to classify text using SVM in C#," 2014. [Online].

[14] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in 10th European Conference on Machine Learning, Chemnitz, 1998.

[15] M. Hu and B. Liu, "Mining Opinion Features in Customer Reviews," Association for the Advancement of Artificial Intelligence, vol. 4, no. 4, 2004.

[16] A. O. D. Community, "OpenNLP Part-of-Speech (POS) Tags: Penn English Treebank," The Apache Software Foundation. [Online].

[17] G. Patil, V. Galande, V. Kekan and K. Dange, "Sentiment Analysis Using Support Vector Machine," International Journal of Innovative Research in Computer and Communication Engineering, vol. 2, no. 1, 2014.