A Survey of Various Machine Learning Techniques for Text Classification

Gaurav S. Chavan[1], Sagar Manjare[2], Parikshit Hegde[3], Amruta Sankhe[4]

Atharva College of Engineering
University Of Mumbai, India

Abstract

Sentiments are expressions of one's words in a sentence. Hence understanding the meaning of text in the sentence is of utmost importance to people of various fields like customer reviews in companies, movie reviews in movies, etc. It may involve huge text data to analyze and it becomes totally unviable for manually understanding the meaning of sentences. Classifier algorithms should be used to classify the various meaning of the sentences. By using pre-defined data to train our classifier and three different algorithms namely Naive Bayes, Support Vector Machines, Decision Trees, we can simplify the task of text classification. Using relevant results and examples we will prove that SVM is one of the better algorithms in providing higher accuracy over the other two algorithms i.e. Naive Bayes and Decision Tree.

Keywords — **Text Classification, Sentiment Analysis, Algorithms, Naive Bayes, SVM, Decision Tree**

I. INTRODUCTION

Today large amount of data is present in online documents over the Internet. As effort to better organize the information of users, researchers have been actively investigating the problem of automatic text classification.[1] Text Classification is now being applied in many contexts, ranging from document indexing based on a controlled vocabulary, to document filtering, automated metadata generation, word sense disambiguation, population of hierarchical catalogues of Web resources, and in general any application requiring document organization or selective and adaptive document dispatching.[2]

Recent years have seen rapid growth in online discussion groups and review sites (e.g., the New York Times' Books web page) where a crucial characteristic of the posted articles is their sentiment or overall opinion towards the subject matter— for example, whether a product review is positive or negative. Labelling these articles with their sentiment would provide succinct summaries to readers; indeed, these labels are part of the appeal and value-add of such sites as www.rottentomatoes.com [1]. Today Text Classification has become the core of Sentiment Analysis and is constantly evolving to become more and more accurate with limited amount of data. Sentiments are nothing but emotions of a person and in what context the emotions are referred.

Today social media platforms like Twitter, Facebook, and MySpace provide people a platform to express their emotions in the digital world; which provide valuable information. But understanding 58 million tweets and 1 billion posts that may generate huge comments in a day is a humongous task in itself. Here sentiment analysis comes into picture, by designing algorithms that are already available and modifying them to suit our needs; this tedious task of understanding the meaning of the sentences can be easily, efficiently and elegantly achieved.

In this paper, we examine and compare the effectiveness of applying machine learning techniques to the sentiment classification problem. A challenging aspect of this problem that seems to distinguish it from traditional topic-based classification is that while topics are often identifiable by keywords alone, sentiment can be expressed in a more subtle manner. [1]

II. SENTIMENTAL ANALYSIS

Definition

Sentiment Analysis is a Natural Language Processing and Information Extraction task that aims to obtain writer's feelings expressed in positive or negative comments, questions and requests, by analyzing a large numbers of documents. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall tonality of a document. [13]

What are the challenges?

Sentiment Analysis approaches aim to extract positive and negative sentiment bearing words from a text and classify the text as positive, negative or else objective if it cannot find any sentiment bearing words. In this respect, it can be thought of as a text categorization task. In text classification there are many classes corresponding to different topics whereas in Sentiment Analysis we have only 3 broad classes i.e. positive, negative and neutral. Thus it seems Sentiment Analysis is easier than text classification which is not quite the case. The general challenges can be summarized as: [13]

1. Implicit Sentiment and Sarcasm
2. Domain Dependency
3. Thwarted Expectations

4. Pragmatics
5. World Knowledge
6. Subjectivity Detection
7. Entity Identification
8. Negation

Hence, it's not easy to do text categorization and understand what the user intends to say (sentiments) because of the above mentioned problems.

III. Approach for the problem

The complexity of the problems varies from high to low. So some problems are easily solvable like World Knowledge and some are difficult like Negation. For this purpose various algorithms like Naive Bayes, SVM and Decision Tree at available at our disposal.

Steps for analyzing the sentiments in the sentence:

1. Firstly we need to decide the classifier algorithms and have an appropriate data for training.
2. Preprocess and label the data.
3. Prepare the data for training.
4. Train the classifier with the help of libraries such as NLTK, libsvm etc.
5. Make predictions by giving new test data to the trained classifier. [17]

IV. Text categorization

Text categorization is the task of assigning a Boolean value to each pair $(d_j , c_i) \in D \times C$, where D is a domain of documents and $C = \{c1 , \ldots , c_{|C|}\}$ is a set of predefined categories. A value of ***T*** assigned to (d_j , c_i) indicates a decision to file d_j under c_i ,while a value of ***F*** indicates a decision not to file d_j under c_i. [2]

In Machine Learning terminology, the classification problem is an activity of supervised learning, since the learning process is "supervised" by the knowledge of the categories and of the training instances that belong to them. [2]

Here we study three supervised classification algorithms namely *Support Vector Machines (SVM)*, *Naive Bayes* and *Decision Tree Learning* and conclude that SVM performs better than the other two in text classification.

V. Machine learning classifiers

Support Vector Machines:

SVM classification algorithms, proposed by Vapnik [3] to solve two-class problems, are based on finding a separation between hyper-planes defined by classes of data, shown in Figure 1. This means that the SVM algorithm can operate even in fairly large feature sets as the goal is to measure the margin of separation of the data rather than matches on features. The SVM is trained using pre-classified documents. [4]

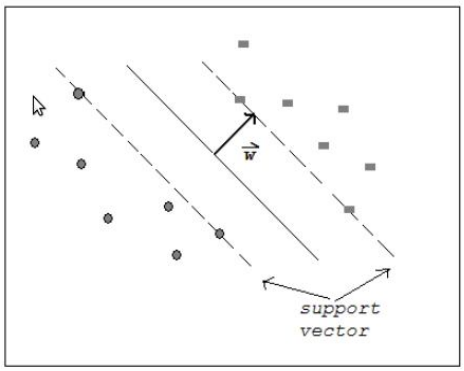

Figure 1: Example of SVM hyper-plane pattern [4]

Naive Bayes Classification:

A Naive Bayes classifier is a well-known and practical probabilistic classifier and has been employed in many applications. It assumes that all attributes (i.e., features) of the examples are independent of each other given the context of the class, i.e., an independence assumption. It has been shown that Naive Bayes under zero-one loss performs surprisingly well in many domains in spite of the independence assumption [5].

$$p(c|d_j) = \frac{p(d_j|c)p(c)}{p(d_j)} = \frac{p(d_j|c)p(c)}{p(d_j|c)p(c) + p(d_j|\bar{c})p(\bar{c})}$$

$$p(c \mid d_j) = \frac{\frac{p(d_j|c)}{p(d_j|\bar{c})} \cdot p(c)}{\frac{p(d_j|c)}{p(d_j|\bar{c})} \cdot p(c) + p(c)}$$

In the context of text classification, the probability that a document $\boldsymbol{d_j}$ belongs to a class ***c*** is calculated by the Bayes' theorem as follows:

Decision Tree Classifier:

Decision Tree Classifier is a simple and widely used classification technique. It applies a straightforward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receives an answer, a follow-up question is asked until a conclusion about the class label of the record is reached. [16]

In this paper we use C4.5 algorithm (Decision Tree). **C4.5** is an algorithm used to generate a decision tree developed by

Ross Quinlan [12]. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. [18]

VI. RELATIVE EFFICIENCY OF SVM AND NAIVE BAYES

Research has shown [14] that SVM scales well and has good performance on large data sets. Linear SVM and Naive Bayes are both highly efficient and are suitable for large text systems. As they are linear classifiers, both require a simple dot product to classify a document. Dumais et al. implies that the linear SVM is faster to train than Naive Bayes [1998]. Training NB is faster since no optimization is required since a single pass over the training set is sufficient to gather word counts. The SVM must read in the training set and then perform a quadratic optimization. This can be done quickly when the number of training examples is small (e.g. < 10000 documents), but can be a bottleneck on larger training sets. Speed can be improved with chunking and by caching kernel values between the training of binary classifiers. [15]

Using the entire vocabulary as the feature set, Rennie and Rifkin found that the SVM algorithm outperformed the Naïve Bayes algorithm used on two data sets; 19,997 news related documents in 20 categories and 9649 industry sector data documents in 105 categories. Naïve Bayes classification algorithms are based on an assumption that the terms used in documents are independent. Both Bayes and SVM algorithms are linear, efficient, and scalable to large document sets [15].

VII. CLASSIFICATION ACCURACY AND WHY SVM IS MOST PREFERRED TEXT CLASSIFIER

Classification accuracy is measured using the average of precision and recall (the so-called breakeven point). Precision is the fraction of retrieved instances that are relevant, while recall (also known as_sensitivity) is the fraction of relevant instances that are retrieved. Table 1 summarizes micro-averaged breakeven performance for 5 different learning algorithms explored by Dumais et al. (1998) for the 10 most frequent categories as well as the overall score for all 118 categories. [9]

	Naive Bayes	Bayes Nets	Trees (C4.5)	Linear SVM
Eam	95.9%	95.8%	97.8%	98.2%
Acq	87.8%	88.3%	89.7%	92.7%
Money-fx	56.6%	58.8%	66.2%	73.9%
Grain	78.8%	81.4%	85.0%	94.2%
Crude	79.5%	79.6%	85.0%	88.3%
Trade	63.9%	69.0%	72.5%	73.5%
Interest	64.9%	71.3%	67.1%	75.8%
Ship	85.4%	84.4%	74.2%	78.0%
Wheat	69.7%	82.7%	92.5%	89.7%
Com	65.3%	76.4%	91.8%	91.1%
Avg Top 10	81.5%	85.0%	88.4%	91.3%
Avg All Cat	75.2%	80.0%	N/A	85.5%

Table 1: Micro-averaged breakeven performance summarization [9]

Linear SVMs were the most accurate method, averaging 91.3% for the 10 most frequent categories and 85.5% over all 118 categories. These results are consistent with Joachims (1998) results in spite of substantial differences in text pre-preprocessing, term weighting, and parameter selection, suggesting the SVM approach is quite robust and generally applicable for text categorization problems. [9]

Figure 2: ROC curve representation. [9]

Figure 2 shows a representative ROC curve for the category "grain". This curve is generated by varying the decision threshold to produce higher precision or higher recall, depending on the task. The advantages of the SVM can be seen over the entire recall-precision space. [8]

Joachims in his experiment compared the performance of SVM with Naive Bayes and C4.5 decision tree learner among others. He used two data sets, first one was "ModApte" split of the Reuters-21578 datasets compiled by David Lewis and the second one was Ohsumed corpus compiled by William Hersh. [8]

Results from his experiment showed that on Reuters data set k-NN performed better than the other conventional methods

and in comparison to conventional methods all SVMs perform better independent of the choice of parameter. [8]

The results for the Ohsumed data show that C4.5 has micro averaged precision/recall breakeven point of 50 which is far lesser than SVM. This happens because heavy over fitting is observed when using more than 500 features. Naive Bayes achieved a performance of 57. Again Polynomial SVM with 65.9 and RBF SVM with 66, we get that SVMs perform better than conventional methods of classification. [8]

VIII. LITERATURE SURVEY

Sr. No.	Year	Paper	Description
1.	1998	Text Categorization with Support Vector Machines: Learning with Many Relevant Features	Describes the use of SVM for learning with text classifier from examples. It analysis the particular properties of learning with text data and identifies why SVM is good for the task.
2.	2002	Thumbs up? Sentiment Classification using Machine Learning Techniques	This paper deals with the problem of classifying documents not by topic, but by overall sentiments like positive, negative or neutral using Naive Bayes, SVM and Maximum Entropy.
3.	2005	Using Appraisal Groups for Sentiment Analysis	It presents a new method for sentiment classification based on extracting and analyzing appraisal groups such as “very good” or “not terribly funny”. An appraisal group is represented as a set of attribute values in several task - independent semantic taxonomies, based on Appraisal Theory.
4.	2005	Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis	It presented a new approach to phrase level sentiment analysis that first determines whether an expression is neutral or polar expressions. With this approach, the system is able to automatically identify the contextual polarity for a large subset of sentiment expressions, achieving results that are significantly better than baseline.
5.	2007	Automatic Sentiment Analysis in Online Text	The paper consider the emotions as a classification task: their feelings can be positive, negative or neutral. A sentiment isn’t always stated in a clear way in the text; it is often represented in subtle, complex ways. Besides direct expression of the user's feelings towards a certain topic, he or she can use a diverse range of other techniques to express his or her emotions.
6.	2008	Opinion Mining and Sentiment Analysis	This paper covers techniques and approaches that promise to directly enable opinion-oriented information-seeking systems
7.	2010	Twitter as a Corpus for Sentiment Analysis and Opinion Mining	It uses data from micro-blogging site like Twitter and shows how to automatically collect a corpus for sentiment analysis and opinion mining purposes. It perform linguistic analysis of the collected corpus and explain discovered phenomena. Using the corpus, it build a sentiment classifier, that is able to determine positive, negative and neutral sentiments for a document.
8.	2011	Lexicon-Based Methods for Sentiment Analysis	The study presents a lexicon-based approach to extracting sentiment from text.
9.	2013	Unsupervised Sentiment Analysis with Emotional Signals	The authors propose to study the problem of unsupervised sentiment analysis with emotional signals. They incorporate the signals into an unsupervised learning frame work for sentiment analysis. In the experiment, they compare the proposed framework with the state-of-the-art methods on two Twitter datasets and empirically evaluate their proposed framework to gain a deep understanding of the effects of emotional signals.
10.	2014	Comparing and Combining Sentiment Analysis Methods	The study aims at presenting comparisons of popular sentiment analysis methods in terms of coverage (i.e., the fraction of messages whose sentiment is identified) and agreement (i.e., the fraction

			of identified sentiments that are in tune with ground truth). It also develops a new method that combines existing approaches, providing the best coverage results and competitive agreement.

IX. CONCLUSION

This paper introduces different machine learning classifiers for text classification. It provides for theoretical and empirical evidence that SVMs is better for text classification over other classifiers. The analysis concludes that SVMs have higher accuracy and can find and adjust automatically to parameter settings. All this makes SVMs a very promising classifier for text classification.

ACKNOWLEDGEMENT

We are extremely grateful to Ms. Amruta Sankhe for her constant guidance and support.

REFERENCES

1. Bo Pang and Lillian Lee, Shivakumar Vaithyanathan. “Thumbs up? Sentiment Classification using Machine Learning Techniques”. Appears in Proc. 2002 Conf. on Empirical Methods in Natural Language Processing (EMNLP)
2. Fabrizio Sebastiani. “Machine Learning in Automated Text Categorization”.
3. Vladimir Vapnik(1995) “Support-Vector Networks. AT&T Bell Labs., Hohndel, NJ 07733, USA.
4. A. Basu, C. Watters, and M. Shepherd(2002).” Support Vector Machines for Text Categorization.Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS’03).”
5. P. Domingos and M. J. Pazzani, “On the Optimality of the Simple Bayesian Classifier under Zero-One Loss,” Machine Learning,vol. 29, nos. 2/3, pp. 103-130, 1997.
6. Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung Hyon Myaeng(2006). “Some Effective Techniques for Naive Bayes Text Classification”. (Knowledge and Data Engineering, IEEE Transactions on volume 18, issue 11, 2006)
7. Fabrice Colas and Pavel Brazdil. “Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks.”
8. Joachims, T. “Text categorization with support vector machines: Learning with many relevant features.” European Conference on Machine Learning (ECML), 1998.
9. Susan Dumais. “Using SVM for text categorization. (Decision Theory and Adaptive Systems Group Microsoft Research)”
10. S. Rasoul Safavian and David Landgrebe."A survey of Decision Tree methodology".
11. Daniela XHEMALI, Christopher J. HINDE and Roger G. STONEIJCSI. International Journal of Computer Science Issues, Vol. 4, No. 1, 2009
12. Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
13. Sentiment Analysis: A Literature Survey by Subhabrata Mukherjee-IIT-Bombay
14. Kwok, J.T-K. (1998)” Automated Text Categorization Using Support Vector Machine.” Proceedings of the International Conference on Neural Information Processing (ICONIP).
15. Rennie, J.D.M. and R. Rifkin. (2001). “Improving Multiclass Text Classification with the Support Vector Machine.”,May 23, 2002
16. http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/lguo/decisionTree.html
17. https://cloud.google.com/prediction/docs/sentiment_analysis
18. http://en.wikipedia.org/wiki/C4.5_algorithm