Sentiment Analysis in Conversation:
A Comparison of Naïve Bayes and Support Vector Machine Models

Sree Keerthi Matta and **Sai Surya Teja Maddikonda**
Northeastern University
{sreematta, smaddikonda}@ccs.neu.edu

Abstract

Sentiment Analysis or Opinion Mining is a well-known technique used for classifying words, senses, texts, documents according to the opinion, or sentiments they express. In this paper we attempt to make a comparison in the application of Naïve Bayes and Support Vector Machine (SVM) classifiers for Sentiment Analysis. In our work, we pre-process the data set and calculate the Term Frequency for each word in the corpus, which will be used for naïve Bayes approach. A lexicon based approach is employed for SVM approach, by making use of SentiWordNet—a lexical resource for opinion mining, which provides features required for the SVM model.

Introduction

The emergence of social media, blogs and e-commerce has given rise to generation of huge data in terms of user ratings, reviews, recommendations and other forms of online expression. Hence it became fundamentally necessary for businesses to keep track of the online opinion of the users about their products and services, to identify new opportunities and manage their reputations. As the process of manually analyzing the online sentiment from thousands of sources is certainly humongous, the idea of automating this process—by filtering out the noise, understanding the conversations, identifying the relevant content and processing it appropriately—has paved the road for Sentiment Analysis. People give their opinions in the form of unstructured format via various online platforms and the main task in Sentiment Analysis is to preprocess the unstructured data to extract opinion, which can be positive, negative or neutral.

Sentiment Analysis is achieved by employing classification methods, which are a combination of Lexicon approach and Machine Learning approaches. Lexicon approach is a dictionary based approach in which a dictionary of words in a specific language serves as a comprehensive data set. It provides several attributes of a word such as its part of speech, the positive, negative and objective sentiment scores, the meaning of the word, etc. Machine Learning techniques are most widely used for the purpose of classification and prediction of the sentiment as either positive or negative sentiment. Machine Learning algorithms are mainly classified as Supervised Learning or Unsupervised Learning approaches. Supervised approach takes a labeled dataset where each training set has its sentiment already assigned. Unsupervised approach takes unlabeled dataset where the sentiment is not expressed.

Sentiment analysis is categorized into 3 different levels which are: document level, sentence level and entity/aspect level. The difference between these levels is discussed as follows. In the document level analysis, the overall opinion expressed in the document is to be identified. In the sentence level analysis, the opinion expressed at the sentence level is to be identified. But in the entity/aspect level analysis, the focus is directly on the opinion itself.

Related Work

Early work in that area includes Turney and Pang who applied different methods for detecting the polarity of product reviews and movie reviews respectively. This work is at the document level. One can also classify a document's polarity on a multi-way scale, which was attempted by Pang and Snyder among others: Pang and Lee expanded the basic task of classifying a movie review as either positive or negative to predict star ratings on either a 3 or a 4 star scale, while Snyder performed an in-depth analysis of restaurant reviews, predicting ratings for various aspects of the given restaurant, such as the food and atmosphere (on a five-star scale). Even though in most statistical classification methods, the neutral class is ignored under the assumption that neutral texts lie near the boundary of the binary classifier, several researchers suggest that, as in every polarity problem, three categories must be identified. Moreover, it can be proven that specific classifiers such as the Max Entropy and the SVMs can benefit from the introduction of a neutral class and improve the overall accuracy of the classification. There are in principle two ways for operating with a neutral class. Either, the algorithm proceeds by first identifying the neutral language, filtering it out and then assessing the rest in terms of positive and negative sentiments, or it builds a three-way classification in one step. This second approach often involves estimating a probability distribution over all categories (e.g. Naive Bayes classifiers as implemented by Python's NLTK kit). Whether and how to use a neutral class depends on the nature of the data: if the data is clearly clustered into neutral, negative and positive language, it makes sense to filter the neutral language out and focus on the polarity between positive and negative sentiments. If, in contrast, the data is mostly neutral with small deviations towards positive and negative affect, this strategy would make it harder to clearly distinguish between the two poles.

A different method for determining sentiment is the use of a scaling system whereby words commonly associated with having a negative, neutral or positive sentiment with them are given an associated number on a -10 to +10 scale (most negative up to most positive). This makes it possible to adjust the sentiment of a given term relative to its environment (usually on the level of the sentence). When a piece of unstructured text is analyzed using natural language processing, each concept in the specified environment is given a score based on the way sentiment words relate to the concept and its associated score. This allows movement to a more sophisticated understanding of sentiment, because it is now possible to adjust the sentiment value of a concept relative to modifications that may surround it. Words, for example, that intensify, relax or negate the sentiment expressed by the concept can affect its score. Alternatively, texts can be given a positive and negative sentiment strength score if the goal is to determine the sentiment in a text rather than the overall polarity and strength of the text.

Recent works, such as Rosa, Rodríguez and Bressan detect sentiment variations in accordance with the user's profile. In sentiment analysis is important to consider different scores for verbs tenses, negative sentences and others, such as in Sentimeter-Br metric. Recent works, such as Rosa, Rodríguez and Bressan detect sentiment variations in accordance with the user's profile.

Subjectivity/Objectivity Identification

This task is commonly defined as classifying a given text (usually a sentence) into one of two classes: objective or subjective. This problem can sometimes be more difficult than polarity classification. The subjectivity of words and phrases may depend on their context and an objective document may contain subjective sentences (e.g., a news article quoting people's opinions). Moreover, as mentioned by Su results are largely dependent on the definition of subjectivity used when annotating texts. However, Pang showed that removing objective sentences from a document before classifying its polarity helped improve performance.

Feature/Aspect-based

It refers to determining the opinions or sentiments expressed on different features or aspects of entities, e.g., of a cell phone, a digital camera, or a bank. A feature or aspect is an attribute or component of an entity, e.g., the screen of a cell phone, the service for a restaurant, or the picture quality of a camera. The advantage of feature-based sentiment analysis is the possibility to capture nuances about objects of interest. Different features can generate different sentiment responses, for example a hotel can have a convenient location, but mediocre food. This problem involves several sub-problems, e.g., identifying relevant entities, extracting their features/aspects, and determining whether an opinion expressed on each feature/aspect is positive, negative or neutral. The automatic identification of features can be performed with syntactic methods or with topic modeling. More detailed discussions about this level of sentiment analysis can be found in Liu's work.

Problem Statement and Proposed Technique

The targeted application for the proposed model in our work is in the field of medical aid technology for people suffering from memory impairment diseases like Alzheimer's. For the purpose of assisting a memory impairment patient in recognizing and talking to other people, a medical aid device driven by our proposed model would give prompts about the friendliness of the person the patient is talking to. More about the applications of the proposed model will be discussed in the conclusion below.

As for the application mentioned above, for calculating the friendliness of a person the patient is conversing to, we need to identify those sentences which are targeted at the patient. For example, in the sentence *"You have a great sense of humor"*, the object the speaker is referring to is the listener ('you'). But in the sentence *"He has a great sense of humor"*, the speaker refers to some third person. Although the remaining words in both the example sentences are the same and convey a positive sentiment, only the sentence that is targeted to the patient will be considered for calculating the friendliness. Hence, we need to perform Sentiment Analysis on all the sentences for which the patient is regarded as the object of. In our work, we aim to perform Sentiment Analysis at sentence level in a conversational text, and attempt to compare the accuracy results of Naïve Bayes and SVM classifiers.

Choosing Dataset

A great challenge pertinent to our supposed application while using the SVM approach, is to find one of the following:

i. A labelled data set of conversations between two people, in which each sentence in the conversation is labelled positive or negative.

ii. A labelled data set of conversational replies targeted towards the first person, i.e., the object of the sentence is the listener but not some other third person, in which each sentence is labelled positive or negative.

Few data sets are available from Internet Relay Chats, Twitter Conversations, movie dialogs, etc., but none of them were fitting into the criteria we envisaged for our use case. Hence we built a decent dataset of labelled conversational replies, in which each sentence or a reply in the conversation is labelled either as positive or negative. In addition to our dataset, we also made use of SentiWordNet, a lexical resource for opinion mining. SentiWordNet provides the positive and negative sentiment scores of many English words, which can be used as features for the SVM approach.

Data Preprocessing

The Data Preprocessing action is crucial for Sentiment Analysis. The steps involved are as follows:

i. The unnecessary punctuation marks are removed from a sentence in test or training data.

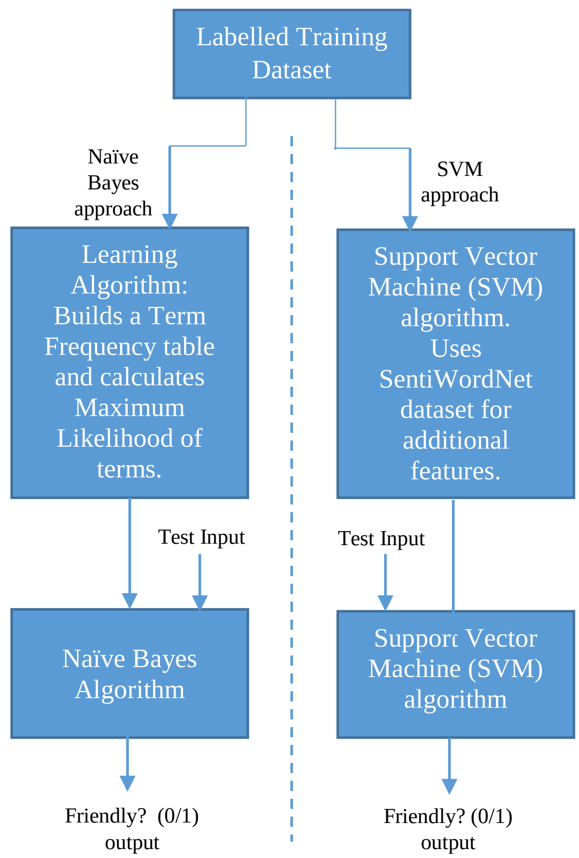
ii. Next the sentence is tokenized into words, thus disregarding the whitespace. Tokenization is the process by which the sentence is broken down into its constituent words.

iii. Now, the stop words are removed from the list of words obtained after the tokenization step. Stop words are those words which have no bearing on the overall sentiment of a sentence. For example, most of the prepositions, interjections and pronouns in English like *he*, *she*, *I*, *it*, *there*, *here*, *these*, *for*, *but*, *of*, *and*, *or*, etc., are stop words.

iv. Also, a spelling correction action is performed on the words at this stage to improve the accuracy of the model. NodeBox English Linguistics API is capable of doing a spellcheck and auto-correction of misspelt words.

v. Finally, the words are stemmed to their root. Stemming is the process by which a word is reduced to its root form. For example, the words *runner*, *running* and *ran* are reduced to its root form *run*.

Workflow



As shown in the workflow block diagram above our labelled training dataset is fed to both the approaches—Naïve Bayes and SVM. Each of the approaches is discussed in detail in the following sections.

Naïve Bayes Approach

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. Using Bayes theorem, the conditional probability of a sentence S and label C_k = friendly/unfriendly, can be decomposed as:

$$P(C_k|S) = \frac{P(C_k)P(S|C_k)}{P(S)}$$

Since the sentence S is made of words $w_1, w_2, w_3, \dots, w_n$ the joint probability mentioned in the numerator $P(C_k)P(S|C_k)$ is equivalent to $P(C_k, S)$.

$$P(C_k, w_1, w_2, w_3, \dots, w_n) = P(w_1, w_2, w_3, \dots, w_n, C_k)$$
$$= P(w_1 \mid w_2, w_3, \dots, w_n, C_k)\, P(w_2, \dots, w_n, C_k)$$

We continue this process till n.

The naïve conditional independence assumes that each feature F_i is conditionally independent of every other feature F_j for $i \neq j$, given a class C.

$$\therefore P(w_i \mid w_{i+1}, \dots, w_n, C_k) = P(w_i \mid C_k)$$

Thus the joint probability can be expressed as

$$P(C_k|w_1, w_2, w_3, \dots, w_n) \propto P(C_k, w_2, w_3, \dots, w_n)$$
$$\therefore P(C_k|w_1, w_2, w_3, \dots, w_n) \propto P(C_k) \prod_{i=1}^{n} P(w_i \mid C_k)$$

Given a training set of n sentences, we construct our frequency distribution table, which provides us all the words with their respective frequency of occurrence in the training dataset, in the friendly and unfriendly sentences classified. Now we predict the friendliness outcome of a new test reply from a person in conversation with the patient using Naïve Bayes as follows: $P(S|C) = \prod_i P(w_i|C)$, where C is the possible class. In our assumption, the class is either Friendly (F) or Unfriendly ($\neg F$). Hence, $P(S|F) = \prod_i P(w_i|F)$ and $P(S|\neg F) = \prod_i P(w_i|\neg F)$

Using the Bayesian result deduced above,

$$P(F|S) = \frac{P(F)}{P(S)} \prod_i P(w_i|F) \qquad \dots (1)$$

$$P(\neg F|S) = \frac{P(\neg F)}{P(S)} \prod_i P(w_i|\neg F) \qquad \dots (2)$$

Dividing (1) by (2)

$$\frac{P(F|S)}{P(\neg F|S)} = \frac{P(F)}{P(\neg F)} \frac{\prod_i P(w_i|F)}{\prod_i P(w_i|\neg F)}$$

Applying Napier logarithm on both sides,

$$\ln\left(\frac{P(F|S)}{P(\neg F|S)}\right) = \ln\left(\frac{P(F)}{P(\neg F)} \frac{\prod_i P(w_i|F)}{\prod_i P(w_i|\neg F)}\right)$$

$$\ln\left(\frac{P(F|S)}{P(\neg F|S)}\right) = \ln\left(\frac{P(F)}{P(\neg F)}\right) + \sum_i \ln \frac{\prod_i P(w_i|F)}{\prod_i P(w_i|\neg F)}$$

Finally, we can classify the reply sentence as follows. If the value $P(F|S) > P(\neg F|S)$, then the sentence is said to convey a positive sentiment and the speaker is friendly to the patient. Else, the speaker can be considered as unfriendly.

SVM Approach

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

In our implementation, the SVM model takes as input two datasets. One is our conversational reply dataset and the other is the SentiWordNet lexical dataset. Our conversational reply dataset contains reply sentences labelled as either 0 (meaning Unfriendly) or 1 (meaning Friendly). SentiWordNet dataset provides positive and negative sentiment scores for many English words. For making the machine learn over the training dataset efficiently, we propose 4 features, which are formulated from both the datasets.

Suppose in a sentence S, we are left with n_p positive sentiment words, and n_n negative sentiment words after data preprocessing, as looked up in SentiWordNet dataset. We define the proportionality of positive words as:

$$P_p = \frac{n_p}{n_p + n_n}$$

Likewise, we define the proportionality of negative words as:

$$P_n = \frac{n_n}{n_p + n_n}$$

Now, we define our 4 features for the SVM model as follows:

i. Proportionality of positive words in the sentence: P_p
ii. Proportionality of negative words in the sentence: P_n
iii. Mean of the positive sentiments scores of the positive words in the sentence: m_p
iv. Mean of the negative sentiments scores of the negative words in the sentence: m_n

The label against each sentence in our training dataset is appended to these 4 features and given to the SVM algorithm. Since the training data is non-linear, we employ a non-linear SVM classifier by applying the Radial Basis Function (RBF) kernel. This allows the algorithm to fit the maximum margin hyperplane in a transformed feature-space. The RBF Kernel on 2 samples x and x', represented as feature vectors in some input space, is defined as:

$$K(x, x') = e^{\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)}$$

We can recognize $\|x - x'\|^2$ as the squared Euclidean distance between the 2 feature vectors. σ is a free parameter. An equivalent, but simpler definition involves a parameter $\gamma = \frac{1}{2\sigma^2}$

$$K(x, x') = e^{(-\gamma\|x - x'\|^2)}$$

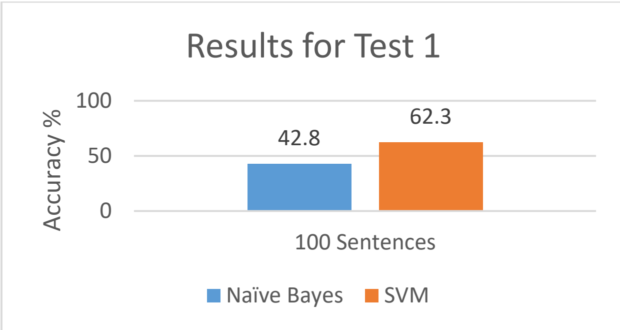
Since the value of the RBF kernel decreases with distance and ranges between zero (in the limit) and one (when x = x'), it has a ready interpretation as a similarity measure.

The complex algorithmic computations involving hyperplane plotting are handled by Scikit-learn—a Machine Learning toolkit in Python, built on NumPy, SciPy and matplotlib Python libraries.

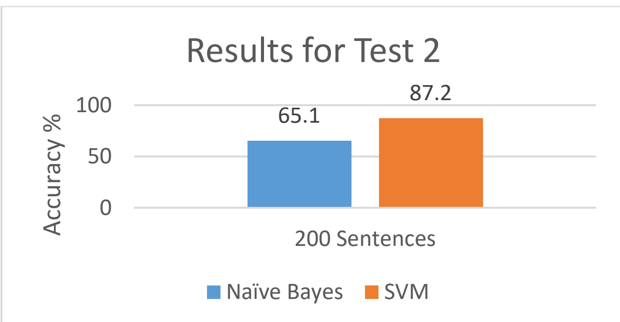
Result Analysis

We experimented with both the approaches by conducting 3 tests, varying the sizes of training datasets and changing the test data sets in each test. The training and test datasets consist of a list conversational reply sentences. For the test dataset input, the labels against each sentence is compared with the friendliness predicts from the models, and then accuracy is calculating.

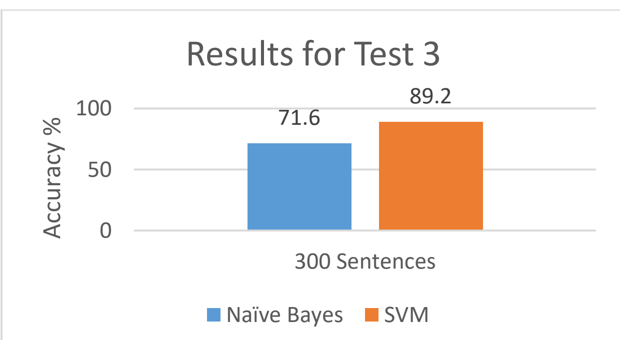
Test 1
Training dataset: 100 sentences
Test dataset: 40 sentences



Test 2
Training dataset: 200 sentences
Test dataset: 40 sentences



Test 3
Training dataset: 300 sentences
Test dataset: 40 sentences



From the results, we propose that the accuracy performance of both the classifiers: Naïve Bayes and SVM will improve with increasing the size of the training dataset. Also, an attempt to make the training dataset more comprehensive, by including positive and negative sentiment sentences encompassing a variety of emotional and linguistic elements will improve the performance of both the approaches.

Conclusion

Our work, where we compared the accuracy performance of Naïve Bayes and SVM approaches for conversation analysis, is targeted towards its application in medical aid devices for assisting people suffering from memory impairment diseases. As mentioned previously in the result analysis, by making the training datasets more comprehensive and by increasing the size of the datasets, there can be a significant improvement in the friendliness prediction and in the accuracy performance.

Future Work

There have been many works in Sentiment Analysis concerning movie reviews, Twitter tweets, financial markets, product reviews, etc. But there are few works so far on conversation analysis pertinent to analyzing the sentiment expressed in chats and conversational sentences. In the future, there is scope to calculate the actual friendliness factor—the percentage by which the speaker is friendly/unfriendly to the patient. Also, there is a possibility to increase the number of features in the SVM approach by considering more complex features like the effect of one word over another in a sentence, the effect of one sentence on the other sentences that follow, the order of sentences in a conversation, etc. With the implementation aforementioned features in SVM approach, there is great likelihood of the accuracy performance increasing.

Acknowledgments

Our sincere thanks to professor Christopher Amato and the teacher assistant Matthew Corsaro for teaching the path to start playing with Artificial Intelligence and basics of Machine

Learning. Special thanks to the creators of SentiWordNet lexical dictionary for opinion mining and Scikit-learn Python Machine Learning Toolbox.

References

Andreevskaia, A.; Ide, N. 2007. Sentiment and Subjectivity Analysis – An Overview. EuroLAN 2007.

Sentiment Analysis. n.d. In *Wikipedia*. Retrieved on December 4, 2016, from https://en.wikipedia.org/wiki/Sentiment_analysis

Turney, Peter 2002. Thumbs Up or Thumbs Down Semantic Orientation Applied to Unsupervised Classification of Reviews. Proceedings of the Association for Computational Linguistics, pages 417–424.

Pang, Bo; Lee, Lillian; Vaithyanathan, Shivakumar. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79–86.

Pang, Bo; Lee, Lillian. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. Proceedings of the Association for Computational Linguistics (ACL), pages 115–124.

Snyder, Benjamin; Barzilay, Regina. 2007. Multiple Aspect Ranking using the Good Grief Algorithm. Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL), pages 300–307.

Vryniotis, Vasilis. 2013. The importance of Neutral Class in Sentiment Analysis.

Koppel, Moshe; Schler, Jonathan. 2006. The Importance of Neutral Examples for Learning Sentiment. Computational Intelligence 22, pages 100–109.

Ribeiro, Filipe Nunes; Araujo, Matheus. 2010. A Benchmark Comparison of State-of-the-Practice Sentiment Analysis Methods. Transactions on Embedded Computing Systems.

Taboada, Maite; Brooke, Julian. 2011. Lexicon-based methods for sentiment analysis. Computational Linguistics. 37 (2): 272–274.

Thelwall, Mike; Buckley, Kevan; Paltoglou, Georgios; Cai, Di; Kappas, Arvid. 2010. Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology. 61 (12): 2544–2558.

Rosa, Renata Lopes; Rodríguez, Demóstenes Zegarra; Bressan, Graça. 2015. Music recommendation system based on user's sentiments extracted from social networks. Transactions on Consumer Electronics. 61.

Rosa, Renata Lopes; Rodríguez, Demóstenes Zegarra; Bressan, Graça. 2013. SentiMeter-Br: A Social Web Analysis Tool to Discover Consumers' Sentiment. IEEE 14th International Conference on Mobile Data Management.

Linguistics Library. n.d. In *NodeBox*. From https://www.nodebox.net/code/index.php/Linguistics

Esuli, Andrea ; Sebastiani, Fabrizio; SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining.