# Literature Review

1. **Sentiment Analysis and Opinion Mining (Bing Liu)**

**Chapter 1**

Prior to the explosion of review blogs, news, social media, when individuals had the intention of buying a product/service, they asked their neighbours, family and friends who have had experience with said product; At this point in time, one is no longer constrained by the reliance on people they know – the internet is full of user reviews and public discussion. There’s however so much information available a human would have difficulty reading through them all and evaluating for the overall sentiment. Hence, an automated means of doing this is necessary. The applications of systems such as these aren’t restricted to the obvious review-based domains of films, products and services but also to politics, social sciences and financial sector. In (Bollen, Mao and Zeng, 2011), Twitter moods were used to predict the stock market. In (Bar-Haim et al., 2011; Feldman et al., 2011), expert investors in microblogs were identified and sentiment analysis of stocks was performed. In (Zhang and Skiena, 2010), blog and news sentiment was used to study trading strategies.

There are three types of sentiment analysis: document-based, sentence based and aspect/entity based. Document-based sentiment analysis focuses on the classification of entire documents. It’s assumed that the document focuses on a single entity. Sentence based focuses on each sentence while aspect based classification goes a bit deeper for finer-grained analysis. It assumes that determining the sentiment without determining its target entity is of little use. There are two types of opinion: regular and comparative. Regular opinion expresses opinion on a single entity while comparative compares two entities.

The most important indicators of sentiment are sentiment words. Positive sentiment is typically conveyed through the use of words such as *good, wonderful, brilliant* while negative sentiment is typically conveyed by the use of words such as *terrible*, *poor, bad, worse.* Phrases and idioms also factor in, e.*g. cost an arm and leg.* A list of sentiment words is called a sentiment lexicon. However useful, a sentiment lexicon isn’t enough because:

* Words can both the positive or negative in different contexts.
* A sentiment word in a sentence may not convey any sentiment. E.g. “*Any good enough tv will do.*”
* Sarcasm.
* Sentences without sentiment words can also imply opinions such as “*I am already looking to replace the car I bought 2 months ago – it no longer works!*”

As an NLP problem, sentiment analysis has to solve problems such as coreference resolution (different ways of representing the same object), negation handling and word sense disambiguation. Sentiment analysis is however a restricted form of NLP as it doesn’t attempt to understand the entire text but to determine some aspects of it.

**Chapter 2**

An opinion can be said to consist of 4 components: (*g, s, h, t*) where g is the opinion target, s is the sentiment, h is the opinion holder and t is the time when the opinion was expressed. The target object can also be referred to as an entity (usually a product, service, topic, issue, person etc. ) which is defined as e: (*T, W*) where *T* is a hierarchy of parts, sub-parts etc and *W* is a set of attributes. Most applications do not need a high level of detail that this definition requires. Hence, the term *aspect* (denote both parts and attributes) is introduced. With the introduction of an aspect, we say that an opinion is defined as where is the name of the entity, is an aspect of , is the sentiment on aspect of entity , is the opinion holder and is the time when the opinion is expressed by . When the opinion is the entity itself, the special aspect GENERAL is used to denote it.

**Chapter 3**

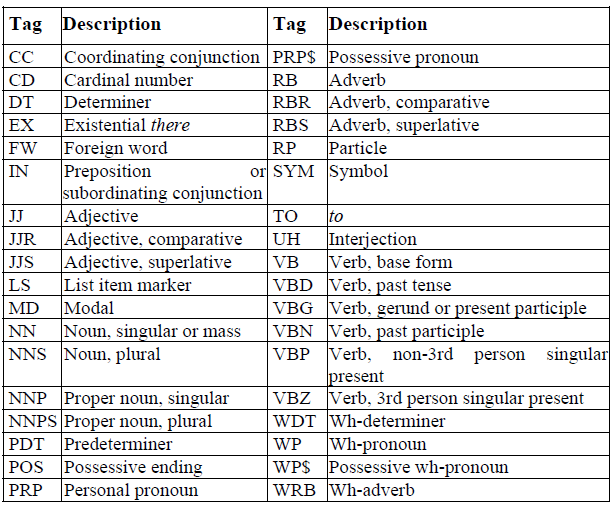
Document sentiment classification assumes that we only evaluating on a single entity:

(\_, GENERAL, s, \_, \_).

As sentiment classification is a text classification problem, we can use existing supervised learning approach such as naïve bayes and support vector machines. Pang, Lee and Vaithyanathan (2002) used this method to classify move reviews using unigrams. Other features:

* Terms and their frequency.
* POS. words of different parts of speech may be treated differently. (POS tags and their n-grtam features can also be used).
* Sentiment words and phrases
* Rules of opinion
* Sentiment shifters – words that change sentiment orientations from positive to negative and vice versa.
* Syntactic dependency.

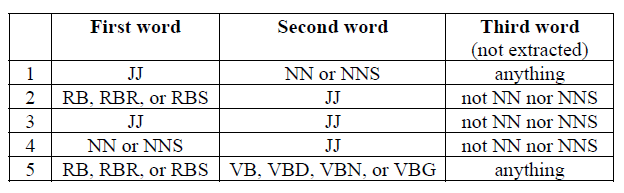
POS tags:



The score function in (Dave, Lawrence and Pennock, 2003) based on words in positive and negative reviews and the aggregation method in (Tong, 2001) using manually compiled domain-specific words and phrases. In (Martineau and Finin, 2009), a new term weighting scheme called Delta TFIDF was proposed. In (Qiu et al.,2009), a lexicon-based and self-supervision approach was used. In (He,2010), labeled features (rather than labeled documents) were exploited for classification. In (Mejova and Srinivasan, 2011) the authors explored various feature definition and selection strategies. In (Bespalov et al., 2011), sentiment classification was performed based on supervised latent n-gram analysis.

Turney, 2002 described a method for unsupervised learning based on POS tags:

Step 1 :



Extract two consecutive words if their POS tags conform to the pattern the above table. Next the sentiment orientation of the extracted phrase is calculated using the PMI measure:

PMI measures the degree of statistical dependence between two terms. Where ) is the co-occurrence probability of and and is the co-occurrence probability of the two terms if they are statistically independent.

Step 2: The SO of a phrase is then determined with the association with the positive reference word “excellent” and the negative reference word “poor”:

Probabilities are calculated by issuing queries to a search engine and collecting the number of hits. A constraint added is to ensure the search engine only accept documents that are within 10 words of each other. Hence we can redefine SO as:

Step 3: Average the SO of all phrases.

84% accuracy for automobile reviews and 66% for movie reviews.

**Chapter 4**

Subjectivity classification simply aims to determine whether or not a sentence is subjective or objective. The first step is to determine whether or not the sentence expresses an opinion or not. The second step is to classify the opinion into positive or negative.

**Chapter 5**

There are two parts of the aspect-based classification problem: aspect extraction and aspect sentiment classification.

Aspect Sentiment Classification: Supervised learning and lexicon-based approach. For supervised learning the approach is to use parsing to determine the dependency and other relevant information. (Jiang et al., 2011), aspect dependent features.

Lexicon based approach (contain sentiment words, phrases, idioms, composite expressions, rules of opinions and the sentence parse tree)

(Ding, Liu and Yu, 2008) approach:

* Mark sentiment words and phrases. Each positive words is assigned a score of +1 and negative words are given a score of -1.
* Apply sentiment shifters. Such as not, never, none, nobody, nowhere, neither, cannot.
* Handle but-clauses
* Aggregate opinions:

Where is a sentiment word/phrase in sentence s, is the distance between aspect and sentiment word in s. is the sentiment score of

Aspect extraction:

* Frequent nouns & noun phrases
* Exploiting opinion and target relations
* Supervised learning
* Topic modelling

1. **Sentiment Analysis of Text Using SVM – ( Yong Yang, Chun Xu, Ze Ren, 2011)**

In their article, <authors> suggest that the process of sentiment analysis can be reduced to examining text as a collection of concept words, thereby removing the need for extracting the relationships between sentences and paragraphs. The document text is then transformed into a vector space model of the form where refers to the term and the refers to the corresponding weight. The weight is calculated using tf-idf which has the formula:

where refers to the frequency of the term in document , is the total number of documents while refers to the number of documents containing the term .

After feature extraction takes place, it’s discovered that there are a large number of words in the feature space, hence dimensionality reduction is necessary. The assumption is that for each category, words that are common across them are less important that word that aren’t. Hence, these words are removed. Afterwards, features are scored based on mutual information, information gain, expected cross entropy and text evidence.

The process by which features are extracted is as follows:

1. Initialise with all words that appear in this category
2. Calculate mutual information by using the formula:

where is the frequency of item t from the training corpus in category , is the number of documents in which the term appears in the category while N is the number of documents in which the term appears in all of the training corpus.

1. The features are sorted according to their mutual information
2. Select the highest features.

During pre-processing, text undergoes word separation, stop word removal and statistic generation.

Using recall and precision as measures of accuracy, the classifier had an average of 81.11% recall in closed test and precision of 81.42% while the open test had an average of 81.09% and precision of 81.12%.

1. **Sentiment Analysis of News Articles for Financial Signal Prediction (Jinjian Zhan, Nicholas Cohen, Anand Atreya)**

The authors attempted to perform classification on news articles by use of a classifier. In order to generate the training and test data, they used a manual method (employed human judgment) as well as an automatic method (by monitoring market movements).

Rather than use machine learning for the sentiment classification, natural language processing was employed instead. Using maximum entropy the Stanford classifier was used for the training and prediction of sentiment based on content from the New York Times articles.

For classification, two methods were employed: manual/human classification an automated classification.

Manual classification involved using a human reader to classify the articles. This is a time-consuming, arduous task and due to the time constraints on the project, the authors were only able to classify two months’ worth of articles. During manual classification, the authors used guidelines such as considering mergers, lower interest rates as good while corruption, lawsuits and rising interest rates were considered unfavourable.

For the extraction of features, article headlines were used as one set of features, article body was used as another and counted sentiment words were set up into bins and these were used as a final feature. In one classification set, only the first two features were used and the classifier achieved F1 scores of 0.581, 0.614 and 0.568 in the positive, neutral and negative classes respectively. When all three sets of features were used, the classifier worsened in accuracy of classifying positive and negative documents with F1 scores of 0.556 and 0.545 while the accuracy for neutral documents increased with an F1 score of 0.701. The reason the authors provided for this behaviour is that the discussion of business issues doesn’t typically employ individual feelings.

Automatic classification involved the use of the movement of the stock market to classify the articles. The dataset used was insufficient for use for a single entity or industry. Hence it was used for the entire S&P 500 index data. The current project will not be suffering from this ailment as the data will be collected using google api hence the application can be as specific/targeted as possible. To do this, the log return of a day’s close is divided by the log return of the previous day’s close. They accounted for the lag between publishing of news articles and the effect on the market. They also accounted for the fact that articles also discuss the previous day’s price. The classifier had F1-scores of 0.269, 0.386 and 0.368 for positive, neutral and negative articles respectively. They attributed this to articles not correlating to the market, poor article selection and suggested that news articles are perhaps better suited to long-term prediction rather than day-to-day prediction. On further work by filtering based on the fact that the articles were from “Financial Desk”, the positive and neutral classes improved by 10% - 20%.

1. **Stock Price Prediction Using Financial News Articles (M. I. Yasef Kaya, M. Elif Karsligil)**

In their paper, <authors> highlighted a way of improving the accuracy of the predicted news articles by using news articles gathered for one year. The news articles are paired with the delta (change in price) prices for each day. Features are then extracted for classification. The most useful features are selected. As with other classification systems, training and then testing follow.

The authors reasoned that in automatic classification of the training and test data, the accuracy (in prior research) tends to be low due to the assumption that the articles affect stock price. However, this assumption isn’t generally true. Hence, the task they reasoned is to select articles only that have an effect on the stock market. A positive article is indicated by a rise in the stock price and vice versa for a decrease in the stock price. Rather than using single words, use of a noun and verb instead was used as they assumed that together these provide more information than a single word. They also omitted stop words such as “by”, “from”.

Even with this omission, the noun-word features are a considerable amount as a single article would have thousands of these couples. Hence, to perform the automatic extraction of useful word couples, the chi-square weight was used:

is the weight of the feature in the class . is the total number of document while is the number of documents in containing .

Afterwards, an SVM was used to perform the classification on a single document. That is, all the articles are aggregated into a single text file. The system had an accuracy of 60% which they claimed is better than randomness (50% accuracy).

1. **Using News Articles to Predict Stock Price Movements(Győző Gidófalvi)**

In his paper, Gyozo attempts to work in line with the efficient market hypothesis which states that profit opportunities in the market are exploited as soon as they arrive. To do this, they use high frequency information retrieval for both stock prices as well as news articles.

A window of influence is defined which is used to evaluate the possible effect of a news article. In addition, news articles that aren’t published within the opening and closing market times are filtered out.

To establish how stable a stock is a β-value is calculated using the linear regression on data-points (Δ index-price, Δ stock-price). Hence, a β-value of 1 means that whenever the index price changes by δ, the stock price is expected to change by δ as well. A β-value of 2 means that whenever the index price changes by δ, the stock price is expected to change by 2δ as well. A β-value of greater than 1 are relatively volatile and the inverse is the case for stocks less than 1.

In order to remove the effects of the exponential change in price, the formula is

The movement of a stock within a time interval is

where is the change in the stock price and is the change in index price. A news article with timestamp can then be measured with offsets to receive a score of .

Movement classes can then be defined from these equations:

Where and are threshold values. Naïve Bayesian can then be used to predict the probability of a document belonging to a class:

Or for each word :

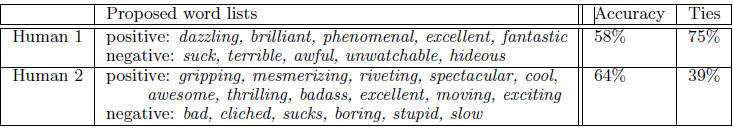
The predictive power of the classification/system discussed is low with the system performing worse than randomness.

The <authors> start by highlighting that news articles are majorly different from reviews or traditional pieces of texts that have had the focus of sentiment analysis research. Another issue they pointed out is that news aren’t necessarily fixed on one target while reviews tend to be on the same concrete object.

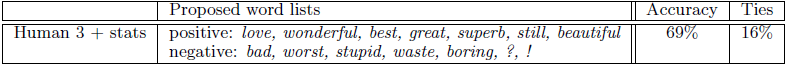
SVMs are an appropriate tool because they are resistant to blog noise, can handle large feature sets consisting of bag of unigram and bigram words feature sets, and are traditionally good at similar tasks like topic based classification.

**Thumbs Up? Sentiment Classification using Machine Learning Techniques (Bo Pang, Lillian Lee and Shivakumar Vaithyanathan)**

The authors tested the hypothesis that humans are relatively good at distinguishing between the positive and negative reviews. They also tested the hypothesis that the use of words that express strong sentiments would be sufficient to classify texts. They asked two students to choose good words that are indicative of positive and negative sentiments. They then simply used their responses to count the number of positive and negative words proposed by the students. The accuracy of the human classifiers is shown below- where ties mean that the two documents were rated equally likely:



In comparison they generate lists of positive and negative words based on frequency counts:



They concluded that rather than relying on intuition, exploring corpus-based techniques is a worthy investment.

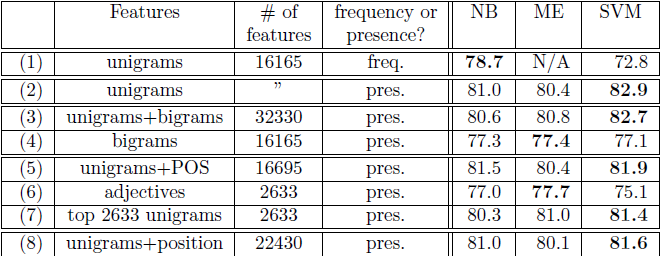
The purpose of their research is to determine whether sentiment classification can be treated as a special case of topic-based classification or whether special/specific sentiment-classification methods are needed.

The method is to define be the set of m features that can appear in a document. Define as the number of times occurs in a document d. Then each document can be represented by .

Naïve Bayes

Class The Naïve Bayes classifier is defined by observing the Bayes rule:

or



**Sentiment Analysis and Applications: A survey(Walaa Medhat, Ahemd Hassan, Hoda Korashy)**