SENTIMENT ANALYSIS FOR MOVIE REVIEWS

SHRAVAN VISHWANATHAN

M.Tech (CSE) Galgotias University, U.P, India

Abstract- Sentiment analysis is becoming one of the most profound research areas for prediction and classification. In Sentiment mining or opinion mining, we basically try to analyse the results or predict outcomes that are based on customer feedback or opinion. Especially applications involving stock prediction, movie and product review analysis. The study involves automated text mining which in turn includes understanding the meaning of the sentence. This can be achieved with the help of appropriate tokenization and thereafter classifying the outcome by comparing the tokens with a specific sentiment dictionary. In this article we take up the challenge of automated movie review analysis. A dataset of a movie is created from the movie review website 'Rotten Tomatoes'. We classify the sentences from the dataset as positive or negative using word stem tokenization and there after measure the sentiment weight.

Index terms- sentiment mining, opinion mining, movie review analysis, tokenization

I. INTRODUCTION

Automated text analysis or sentiment analysis is used in fields where products and services are reviewed by customers and critics, by means of social networking sites and blogs. One more application is also in the field of politics where politicians can come to know about the sentiment of the people regarding certain policies and rallies. It helps while decision making and prediction procedures. The crux behind analysing and understanding the customer's feedback and requirements is to analyse the sentiment of public opinion. This technique helps companies and organizations to know the extent to which a product or service is actually accepted in the particular segment. This in turn would help companies re-strategize their particular products or services or help policy makers re-visit their policies again.

Movie review analysis is one of the most popular fields to analyse public sentiment. For our experiment we have taken a random movie dataset from the website 'Rotten Tomatoes'.

Our goal to study how public mood influences the overall movie review, we need reliable, scalable assessments of the public mood, appropriate for practical movie review analysis. Large surveys of public mood over representative samples of the population are generally expensive and time-consuming. Some have therefore proposed indirect assessment of public mood or sentiment [1] [2]. The accuracy of these methods is however limited by the low degree to which the chosen indicators are expected to be correlated with public mood.

We calculate the sentiment of each sentence using word stem tokenization. The sentences once split in the form of tokens are compared with an exhaustive positive, negative word dictionary. A sentiment value is calculated for each sentence and is classified in a sentiment class based on the majority of positive, negative or neutral words. A sentiment can be positive, neutral or negative. The sentiment of the movie review is calculated by the auto summation of the sentiment values. The results are quite encouraging and can be nearly accurate by averaging all sentiments.

II. TOKENIZATION

Tokenization is the process of breaking a stream of text into meaningful words (stems), phrases or symbols [3] [4]. The tokens can be used further for parsing (syntactic analysis) or text mining. Tokenization is generally considered easy relative to other tasks in text mining and also one of the uninteresting phases. However, errors made in this phase will propagate into later phases and cause problems.

The first step in majority of text processing applications is to segment text into words. In English and other European languages, word tokens are delimited by a blank space. Thus, for such languages, which are called segmented languages token boundary identification is a somewhat trivial task since the majority of tokens are bound by explicit separators like spaces and punctuation. A simple program which replaces white spaces with word boundaries and cuts off leading and trailing quotation marks, parentheses and punctuation produces an acceptable performance.

The next step is to handle abbreviations. In English and other European languages even though a period is directly attached to the previous word, it is usually a separate token which signals the termination of the sentence. However, when a period follows an abbreviation it is an integral part of this abbreviation and should be tokenized together with it. The Dr. is pleased. Now, if we ignore addressing the challenge posed by abbreviation, this line would be delimited into The is pleased.

Universally accepted standards for many abbreviations and acronyms do not yet exist. The most common approach to the recognition of abbreviations is to maintain a list of already known abbreviations. Thus during tokenization a word with a trailing period can be looked up in such a list and, if it is found there, it is tokenized as a single token, else the period is tokenized as a separate token.

The third step is segmentation of hyphenated words which answers the question One or two words? Hyphenated segments can cause ambiguity for a tokenizer. Sometimes a hyphen is part of a token, i.e. self-assessment, G-45, thirty-five and sometimes it is not e.g. New Delhi-based. Tokenization of hyphenated words is generally task dependent. For instance, part-of-speech taggers usually treat hyphenated words as a single syntactic unit and therefore prefer them to be tokenized as single tokens.

III.EXPERIMENT AND RESULT

We created a random dataset of the movie 'Captain Philips' (released in 2013) from the popular movie review website 'Rotten Tomatoes' (RT). The dataset holds only text type attribute consisting of 400 examples. Each example is a full review for the movie given by a specific critic. An example of one of the records of the dataset is "This is acting of the highest order in a movie that raises the bar on what a true-life action thriller can do."

Initially when the sentence is fed as input to the process design, it is tokenized as non-letters i.e the text is segmented into separate meaningful words. This process of tokenization may also be referred as stemming where the affixes of the word are clipped from the word to make it concise with minimum length, having the same meaning. The clipping of affixes is handled by a stem porter. Next, the sentence termination is identified which can be achieved by the filter stopword function. This function filters English stopwords from a document by removing every token which matches a word from a built-in stopword list. Stopwords are words that are not critically necessary to the sentence or opinion. Considering the example quoted above the possible stopwords filtered are 'This is', 'of the', 'in a', 'what a', 'on what', 'can do'. However, every token should represent a single English word only.

The next stage is to compare the tokens (example set) with a preloaded dictionary word list. The example set is the set of words which are already tokenized from the original dataset. The set of tokens is compared with a list of positive word dictionary. Here, the comparisons are performed on the basis of term occurrence. This would indicate that an opinion would be weighted as positive, based on the number of positive terms found in the document. We generate a new attributes at this stage called the 'Positive occurrence'. This attribute contains the sum of the number of positive terms in the text review compared with a positive term dictionary.

The same procedure is repeated for the negative term occurrence in the text simultaneously. The tokens are again compared and matched with a negative term dictionary preloaded in the memory. Similarly, an attribute is generated named 'Negative occurrence' which consists of the sum of negative term occurrences in each text review. The block diagram of the methodology discussed is depicted as follows

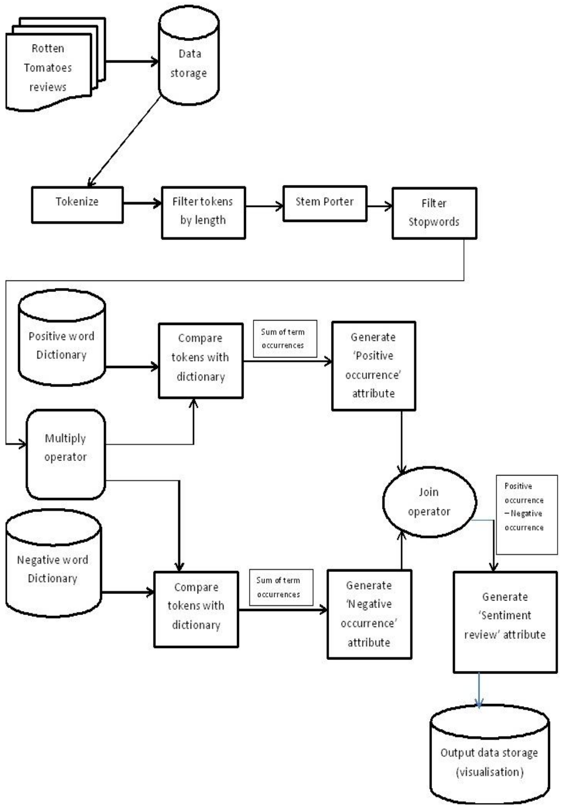
Rotten Tomatoes reviews
Data storage
Tokenize
Filter tokens by length
Stem Porter
Filter Stopwords
Positive word Dictionary
Compare tokens with dictionary
Sum of term occurrences
Generate 'Positive occurrence' attribute
Multiply operator
Join operator
Positive occurrence – Negative occurrence
Negative word Dictionary
Compare tokens with dictionary
Sum of term occurrences
Generate 'Negative occurrence' attribute
Generate 'Sentiment review' attribute
Output data storage (visualisation)

Fig. 1: Flowchart of the discussed tokenization methodology.

The new calculated dataset discussed above is aggregated and joined accordingly. An attribute named 'Sentiment value' is generated at this stage. Sentiment value is calculated by subtracting the 'Positive occurrence' and 'Negative occurrence' attributes (Sentiment review = Positive occurrence – Negative occurrence). However, the sentiment value is calculated by only considering the average of both attribute values.

The value bounds or weights of the sentiment are set to -2.583 to 2.583. The negative value represents a very negative review. As the value nears 0 the review tends to be negative and neutral. The values greater than 0 tend to be positive and very positive. The result for text reviews falling into multiple sentiment classes are depicted below.

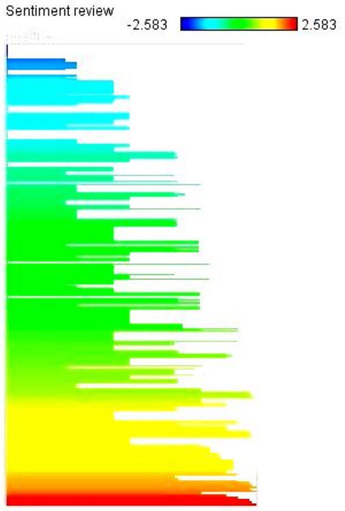

Fig. 2: Multicolour survey graph depicting the varied sentiment classes of different text reviews.

The technique of evaluating the sentiment of a text through tokenization as discussed in this article gives a fair idea of the possible outcome or trend. In this case our motive was to evaluate the sentiment of various critics about the movie ‘Captain Philips’. According to the graph projected we can make out that the review for movie is nearly positive.

Our review provides us with an accuracy of 69.3%. This result is acceptable in order to give an overview of the trend and possible outcomes. However the accuracy of this technique wholly depends on the performance of the tokenizer and the exhaustive positive and negative word dictionary. The efficiency of the tokenizer would affect the number of correct comparisons with an appropriate dictionary.

CONCLUSIONS

Sentiment analysis through text mining is one of the most interesting areas in opinion mining. It opens a whole new set of techniques and methodology for automated classification of sentiments by analysing the semantic and philosophical aspects of a particular string of text. We have proposed a technique of tokenization for classifying text into a number of sentiments. The average performance is fairly acceptable and provides us with a peripheral outlook of the overall trend. In this article we have performed a review analysis of a movie by classifying the texts into sentiments and calculating the overall performance of the proposed methodology. Using this technique we can automatically predict the average review of a movie to be positive, negative or neutral. This is an initial effort to analyse the sentiments from a string of text. In the future we propose to design and model the sematic, syntactic and philosophical association between phrases in the same text string to result in higher classification and prediction accuracy of a sentiment.

REFERENCES

[1] Bollen, Johan, Huina Mao, and Alberto Pepe. "Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena." ICWSM. 2011.

[2] Rahn, Wendy M. "Affect as information: The role of public mood in political reasoning." Elements of reason: Cognition, choice, and the bounds of rationality (2000): 130-50.

[3] Harish, R., et al. "Lexical analysis-a brief study."

[4] Stamatatos, Efstathios. "A survey of modern authorship attribution methods."Journal of the American Society for information Science and Technology 60.3 (2009): 538-556.

★★★