BAYESIAN BASED SENTIMENT ANAYSIS OF TWITTER FEEDS

[1]Jaiyashri Prakash, jai.pk24@gmail.com
[2]George Mathew, george.meg91@gmail.com

ABSTRACT:
In this paper we propose to scale the supervised learning approach of sentiment analysis of twitter data to huge volumes of training data. We achieve the same using MapReduce, an aggregation framework proposed by Google. The solution helps to achieve higher accuracy and we arrive at the best case of evaluation when Naïve Bayes learning is used and the technology can be escalated into real world analysis of Twitter feeds. The usage of sentiment analysis in recommendation engines is also discussed.

KEYWORDS:
Big data analytics, MapReduce, Naïve Bayes supervised learning, sentiment analysis, twitter.

INTRODUCTION:
With the growing popularity of social media channels, huge volumes of data are being produced. The challenge comes in accessing that data and transforming it into something that is usable, meaningful and actionable. One approach to such analytics is sentiment-analysis. Sentiment analysis from social websites provides reliable feedback on public views about any theme of discussion. It has marked its relevance in various domains such as consumer analytics, political analytics, trend mining, brand reputation, etc.

MOTIVATION AND CHALLENGE:

The core of the analytics is split into phases:

- Extracting data from various sources
- Filtering the data according to suit relevance (Preprocessing)
- Synthesizing into statistical figures (Learning)
- Concluding intelligent reports on the target analysis. (Visualizing)

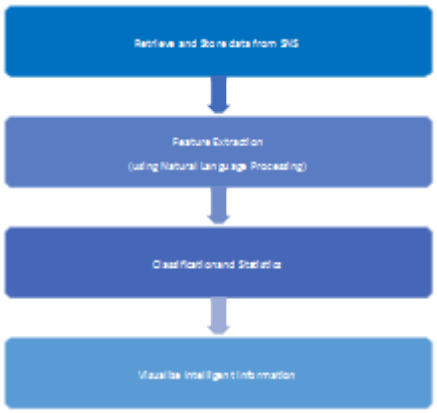

Fig. 1. Core Steps in Analytics.

The challenge lies in not just leveraging across large volumes of data but in arriving at a fast and optimized approach at each phase of the analytics. This is easy said than done. In the context of sentiment analysis we face the following challenges:

1. Handling data from different sources, format, content and size.
2. Parsing the feeds bearing misspelt words, incorrect grammar, abbreviated "sms" slang, emoticons, etc.
3. Handling data having media specific styling such as tags, links, quotes, etc.
4. Arriving at high accuracy while learning from large data.
5. Recognizing new events being fired.
6. Handling sarcasm and ambiguous sentences.

CONTEXT:
Knowing the challenges involved in sentiment analysis, we try to narrow our focus to the context of social networking sites(SNS). Given large volumes of public feeds from users' in SNS we mine consumer's collective opinion revolving around a particular theme of discussion (E.g.: Movies, Products, Current Affairs, Social activities/concerns, Politics, Sports, Entertainment, etc.).

[1]M.Tech at Amrita School of Engineering, Ettimadai, Coimbatore.
[2]Software Engineer III, Payoda Technologies, Coimbatore.

APPROACH:
Sentiment analysis can be handled in two ways: Supervised and Unsupervised learning.

Supervised learning takes an annotated training dataset that is subjected to a classifier for learning. Upon learning, the classifier saves its model based on the algorithm being used. Validation can be done on this classifier model that detects how accurate it is when a new entry (not available in the training dataset) comes in. This saved model is used for testing future input data which may/may not be annotated. The key important task in supervised learning algorithms is the feature extraction phase. In sentiment analysis, commonly used feature extraction techniques include: term presence and their frequency, part-of-speech information, negations, opinion words and phrases. Examples of supervised learning algorithms include Naïve Bayes, Maximum entropy method (MaxENT), Support vector machines (SVM). Although supervised learning bears the advantage of having better performance than unsupervised learning (especially in opinion mining), the strength of the classification is dependent on the amount and quality of the training data used. It may fail when training data is insufficient and acquisition of large amounts of training data is expensive.

Unsupervised learning takes unlabeled data and classifies by comparing the features of the given text against word lexicon whose sentiment values are determined prior to their use. Clustering (k-means, density-based, hierarchical), Self-organizing maps (SOM), adaptive resonance theory (ART) are the commonly used unsupervised algorithm. For example, a prominent work on unsupervised sentiment analysis was done by [Turney, 2002] where point-wise mutual information (PMI) was calculated between the tokens of the review versus the cluster centers of "poor" and "excellent". The sentiment of a document is calculated by the average semantic orientation of all such phrases. Accuracy of 66% was achieved using this technique over the movie review domain. Feature extraction in unsupervised learning are done by dimensionality-reduction techniques such as principal component analysis, independent component analysis, singular-value decomposition, etc. The complexity of unsupervised algorithm and lack of scalability makes us prefer supervised techniques for opinion mining.

PROPOSED MODEL:

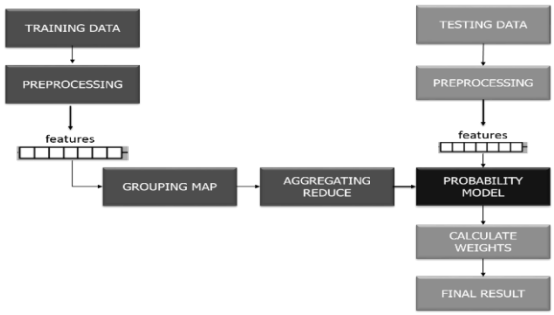

Fig. 2. Flow chart for testing the sentiment of a tweet.

From the literature survey we can infer that most of the experiments have considered emoticons as a noise and there is no effect on the classifier when the testing data with emoticons are introduced. Another observation is, the sentiments that define the entire phrase are adjectives supported by the verbs and adverbs. In order to bypass the naïve approach of tokenizing words and removing irrelevant words (stop words, emoticons, punctuations, etc.) we can use a part-of-speech tagger that tags each word in a phrase with its corresponding part-of-speech (noun, verb, adjective, etc.). Through this, we retrieve only adjectives, adverbs and verbs and polarize them for a given sentence. It is also evident that large volumes of data have not been used for learning. Accuracy of the classifier would increase if this was handled.

Hence for the research scope, the challenges we want to focus are:
1. How to have quick and efficient learning over large volumes of data?
2. In the context of twitter, how do we exploit emoticons for polarizing sentiments?

Our approach is a "semi-supervised" approach that uses Naïve Bayes to deduce a probability count of the tokens in the entire corpus. We use an exhaustive training dataset having 15 lakh annotated twitter feeds. MapReduce is used to speed up the whole classification process. We use regular expressions to tokenize emoticons, negation words, and handle lengthening or words. Different cases of the feature input space was considered when parsed with Naïve Bayes:

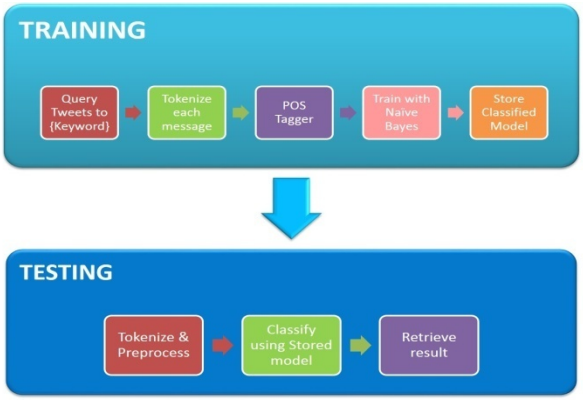

Fig. 3. Training and testing cycle of a tweet.

1. Unigrams: Adjectives, Adverbs and Verbs
2. [1] + Negations + Emoticons
3. [2] + lengthening of words
4. Unigrams + Bigrams + Emoticons (w/o stop word)
5. Unigrams + Bigrams + Emoticons (with stop words)

Validation is done over 10% of the training dataset and accuracy, precision and recall are calculated over the confusion matrix hence generated.

TECHNICAL SPECIFICATIONS:

The implementation of the model was done on JAVA primarily due to its API support and development ease.
A sample training set was taken from www.thinknook.com which provided an annotated dataset consisting of 1.578 million tweets consisting of 0.788 million positively classified tweets and 0.79 million negatively classified tweets.

Each tweet is parsed to obtain its adverbs, adjectives and verbs using a Parts Of Speech (POS) Tagger and its corresponding sentiment, 1 for a positive sentiment and 0 for its negative sentiment. So each word is independently stored in the database along with its corresponding sentiment in that context. After in depth analysis we chose mongoDB for storing data due to its high read speeds and its inbuilt support for map-reduce for superior computational ability.

Post storage each word's sentiment probability is computed using a Bayesian approach for statistics computation. Every word is effectively looked up in the database (map) and a corresponding aggregation of its sentiment is performed (reduce).For example in the entire corpora the word "good" occurs 1000 times, 900 times in a positive scenario and 100 times in a negative scenario. We can effectively say that the positive sentiment probability of "good" is 0.9 and the negative sentiment probability is 0.1. Eventually, each word has a positive sentiment probability and a negative sentiment probability associated with it.

Similarly the emoticons are also parsed using regular expressions and we can extend this approach to bigrams too. Now we have obtained a trained reference for the corpora.

We can then implement over testing data (10% of the training set). Each tweet is then taken and parsed using the same POS tagger and an aggregated weight of positive and negative sentiments is computed. We can then effectively classify the tweet as a positive or a negative one based on the obtained weight.

Real time tweet analysis can also be done either using the tweet4j API provided by Twitter Developer Central to obtain the tweets based on required search parameters.

EXPERIMENTAL RESULTS:

The results below depict the accuracy, precision and recall over the various cases. The best case is seen in the case where we train unigrams, bigrams and emoticons without considering negation words independently. We can understand the impact of emoticons for sentiment analysis and the speed of the algorithm is increased with the use of regular expressions to parse emoticons rather than a dictionary approach.

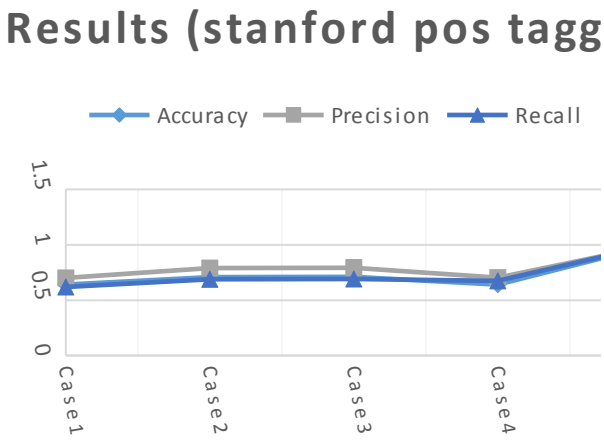

Fig. 4. Results obtained.

CONCLUSION:

Based on the experimental results we conclude that the feature space and preprocessing steps has an impact on the learning algorithm being used. For the context of twitter data large volumes of training data is vital to cope with the vast dynamics.

SCOPE FOR IMPROVAL:

The approach can be further extended by using the following techniques:

1. Using trigrams and higher order n-grams for greater accuracy.
2. Adopting unsupervised methods to make the system smarter.
3. Adopting other supervised techniques like Support Vector Machines and Neural Networks for a greater efficiency by keeping the performance toll in consideration.

ACKNOWLEDGEMENT:

We express our gratitude towards Sathakkathullah Abdul Kafar (Sathak) for giving us the opportunity and finding us eligible to work on this idea. I would also like to thank Ravikumar Kuppusamy (Ravi) for his able support and unbiased opinions during this endeavour. Most of all we would like to thank each other in being able support and greatest judge in times thick and thin.

REFERENCES:

[1] Kouloumpis, E., Wilson, T., & Moore, J. "Twitter Sentiment Analysis: The Good the Bad and the OMG!" Fifth International AAAI Conference on Weblogs and Social Media, ICWSM 2011, July 17-21, 2011, Proceedings, (S. 538-541).

[2] Thomas Lake, "Twitter Sentiment Analysis", April 2011.

[3] Saif, H., He, Y., & Alani, H. "Semantic Sentiment Analysis of Twitter", 11th International Semantic Web Conference, November 2012.

[4] Go, A., Bhayani, R., & Huang, L. "Twitter Sentiment Classification using Distant Supervision" Processing, 2009.

[5]James Spencer and Gulden Uchyigit, "Sentimentor: Sentiment Analysis of Twitter Data", Proceedings of the First International Workshop on Sentiment Discovery from Affective Data (SDAD 2012).

[6] Pritam Gundecha, Huan Liu, "Mining the Social Media: A Brief Introduction", Tutorials in Operational Research Informs, 2012.

[7] Wilas Chamlertwat, Pattarasinee Bhattarakosol, Tippakorn Rungkasiri,

"Discovering Consumer Insight from Twitter via Sentiment Analysis", Journal of Universal Computer Science, vol. 18, no. 8, 2012.

[8] Nagy, A., & Stammberger, J. "Crowd Sentiment Detection during Disasters and Crisis." Proceedings of the 9th International ISCRAM Conference, (S. 1-9). Vancouver, Canada, 2012

[9] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. "Sentiment analysis of Twitter data." Proceedings of the Workshop on Languages in Social Media (S. 30--38). Stroudsburg, PA, USA: Association for Computational Linguistics. 2011

[10] Barbosa, L., & Feng, J. "Robust sentiment detection on Twitter from biased and noisy data." Proceedings of the 23rd International Conference on Computational Linguistics: Posters (S. 36--44). Stroudsburg, PA, USA: Association for Computational Linguistics, 2010.

[11] Jiang, L., Yu, M., Zhou, M., Liu, X., & Zhao, T. "Target-dependent Twitter Sentiment Classification." Computational Linguistics, 151-160, 2011.

[12] Pang, B., & Lee, L. "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts." Proceedings of ACL, (S. 271-278), 2004.

[13] Pang, B., & Lee, L. "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales." Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, (S. 115-124). Ann Arbor, Michigan, USA, 2005.

[14] Pang, B., & Lee, L. "Opinion Mining and Sentiment Analysis. Foundations and Trends" in Information Retrieval, 2006

[15] Pang, B., Lee, L., & Vaithyanathan, S. "Thumbs up? Sentiment classification using machine learning techniques." 7th Conference on Empirical Methods in Natural Language Processing, (S. 79-86). Philadelphia, Pennsylvania, USA, 2002

[16] Saif, H., He, Y., & Alani, H. "Alleviating Data Sparsity for Twitter Sentiment Analysis." Workshop: The 2nd Workshop on Making Sense of Micro posts, 2012.

[17] Peter D. Turney, "Thumbs Up or Thumbs Down? Semantic orientation applied to unsupervised classification of reviews", National Research Council Public archives, Canada, 2002.

[18] Saifee Vohra, et. al., "Applications and Challenges for Sentiment analysis: A survey", IJERT, Vol.2 Issue.2, February 2013.

[19] Vallikannu Ramanathan, T. Meyyapan, "A survey of text mining", International conference on technology of business management, March 2013.