International Journal of Current Advanced Research
ISSN: O: 2319-6475, ISSN: P: 2319-6505, Impact Factor: 6.614
Available Online at www.journalijcar.org
Volume 7; Issue 5(I); May 2018; Page No. 12843-12845
DOI: http://dx.doi.org/10.24327/ijcar.2018.12845.2272

Research Article

THEORETICAL STUDY OF SENTIMENT ANALYSIS USING SUPPORT VECTOR MACHINE

Yash Pankhania*., Sumedh Shewale., Nisha Vanjari and Akshay Wakte

Department of Computer, K. J. Somaiya Institute of Engineering and Information Technology,
Mumbai, Maharashtra

ARTICLE INFO

Article History:

Received 9th February, 2018
Received in revised form 26th
March, 2018 Accepted 17th April, 2018
Published online 28th May, 2018

Key words:

Support vector machine (SVM), TF-IDF, Opinion Mining

ABSTRACT

This work involves the study of Sentiment Analysis with the help of Support Vector Machine. The goal of this study is to understand processes involved in machine learning for sentiment analysis of any given data set. Based on the content of any given text, we are looking to classify the set of words as being positive or negative which defines the text's overall sentiment, opinion, or appraisal about an element or facet of the element from an opinion holder. By studying features to categorize the content of a given text, we learn supervised learning techniques provided by support vector machines.

INTRODUCTION

Sentiment analysis is the process of extracting and analyzing the given data to determine the extent of positivity and negativity present in the expressed opinion [2]. The purpose of this study is to help understand the working of support vector machine to perform sentiment analysis. The study describes all the steps involved in the analysis process.

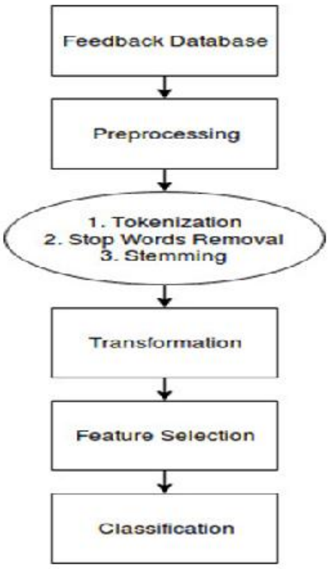

Figure 1 Steps and techniques involved in sentiment classification [6]

The first step involves providing input dataset to the classifier which is required so as to pass on to further steps.

Pre-processing

Pre-processing is a necessary data preparing and cleaning method implemented upon the input dataset subjected for analysis. Pre-processing the data properly helps to reduce the noise in the text which in turn helps improve the performance of the classifier and speed up the classification process, thus aiding in real time sentiment analysis [6]. This step performs following processes-

Tokenization: By providing input in form of sequence of words, tokenization is a task of dividing it up into pieces called tokens as well as removing punctuation marks which serve no use for computing the required result. The notion of a token must first be defined before computational processing can proceed [13].

Stop word removal: A stop-list is basically a list containing a set of stop words. It varies from language to language, as different languages contain different stop words. Any natural language processing system usually contain a range of stop-lists, which depends upon the languages it usually works on, or it might contain a single stop-list that is multilingual. Some of the stop words used in English language are- "and", "or", "but", "the", "it", "an", "nor" etc. they are extremely common words which would appear to be of little value in helping define any users vocabulary entirely [15].

C Stemming: Stemming is a method used to identify the root/stem of a word[9]. Stemming programs/algorithms are usually known as stemmers. The purpose of this method is to remove various suffixes, to reduce the number of words, to

Corresponding author:* **Yash Pankhania
Department of Computer, K. J. Somaiya Institute of Engineering and Information Technology, Mumbai, Maharashtra

have accurately matching stems, to save time and memory space [9]. A simple stemmer looks up the inflected form in a lookup table [6], this method is simple and fast as it only looks into the predefined table for its root form, one of its bigger disadvantage is that all implemented forms must be specifically listed in the table. Eg. “funniest”, “funnier”, “fun” are reduced to the stem “fun”.

Transformation

The weight of every word within the corpus is calculated with the assistance of TF-IDF, so it's straightforward to work out what words within the corpus of documents could be additionally favourable to use during further processes [6]. TF-IDF calculates [7] values for each word in a document defined as below –

yx = fy,x*log(|X|fy,X) [7]

X is a collection of documents, y represents words, x is individual document belongs to X,|X| is the size of the corpus, fy,x is the number of times y appears in x, fy, X is the number of documents in which y occurs in X [7].

Feature Selection

Feature selection is a method that makes classification a lot more economical by reducing the quantity of information to be processed furthermore identifying important factors to be considered in the classification process.

Classification

The main purpose of classification of text is to classify data into predefined classes. The predefined classes here are Positive and Negative classes. Classification is a supervised learning problem. The first course of action in classification is transforming a string based document into a format suitable for the learning algorithm classification task. Information retrieval shows that word stem works well in the form of representation unit. This results in attributed value representation of text. Each word corresponds to a feature which stores the number of times a word occurs in a document as its value. Stop words do not count as a feature (like "and", "or", etc). Performance can be improved by scaling the dimension of the feature with inverse document frequency (IDF) [8].

METHOD

The algorithm is illustrated with a schematic example below. Here, the objects belong to two classes - Positive or Negative.

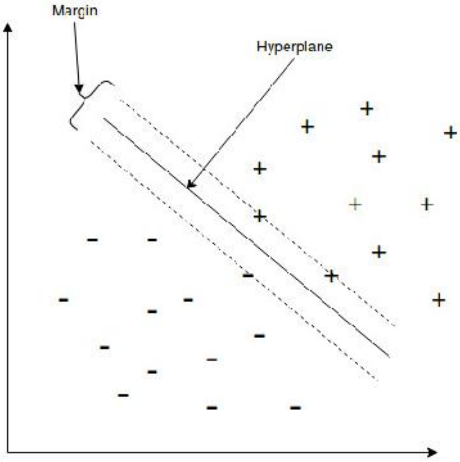

Figure 2 Classification by SVM

There is a separating line which defines a boundary with all the Positive objects on the right side and all the Negative objects on the left side. This boundary is referred to as the ‘Hyperplane’ [17]. A new object falling to the right side of the line is labelled as Positive (or Negative if it falls to the left side of the line).

Applications

The applications for sentiment analysis are endless. More and more we’re seeing it used in social media monitoring to track customer reviews, survey responses, etc. However, it is also used in business analytics and situations in which text needs to be analyzed.

When applied to social media, it can be used to identify spikes in sentiment, thereby allowing you to spot potential product advocates or social media influencers [17]. It can be used to identify when potential negative threads are emerging online regarding any entity, thereby allowing the respective authority to be proactive in dealing with it more quickly. Sentiment analysis could also be applied to any corporate network, for example, by applying it to the email server, emails could be monitored for their general “opinion”. For example, “Tone Detector” is an Outlook Add-in that determines the “tone” of your email as you type [17]. Like an emotional spell checker for all of your outgoing email [17].

In other implementations of Sentiment analysis it can be used for monitoring critical information about earthquake locations and magnitude, riot locations; the monitoring helps policy makers to minimize damage in areas which are expected to be affected next by such events[12]. Another important application of sentiment analysis is the monitoring of the opinions that people submit about pending policy or government-regulation proposals [10, 11].

CONCLUSION

In this paper, we discussed methods and techniques used for sentiment analysis of any text data with the help of Support Vector Machine. We studied text categorization using SVM which can be used to find the polarity of the given text with the help of Hyperplane classification. Hence, we understand that SVM acknowledges some properties of text like High Dimensional feature space and Sparse Instance Vector. Thus as suggested by authors in [13] SVM eliminates the need for feature selection due to its ability to generalize high dimensional feature space.

References

1. Fang Luo, Cheng Li, Zehui Cao: ‘Affective-feature-based Sentiment Analysis using SVM Classifier’, 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)
2. D V Nagarjuna Devi, Chinta Kishore Kumar, Siriki Prasad: ‘A Feature Based Approach for Sentiment Analysis by Using Support Vector Machine’, 2016 IEEE 6th International Conference on Advanced Computing.R. Nicole, “Title of paper with only first word capitalized,” *J. Name Stand. Abbrev*., in press.
3. Ms K. Nirmala Devi, Ms K. Mouthami, Dr V. Murali Bhaskaran ‘Sentiment Analysis and Classification Based on Textual Reviews’, 2012.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

4. Pang, B., & Lee, L., ‘A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts’, In Proceedings of the association for computational linguistics, pp. 271–278, 2004.
5. Theresa Wilson, Janyce Wiebe, Paul Hoffmann, ‘Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis’, In Proceedings Of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), Vancouver, pp. 347–354 2004.
6. Ms Gaurangi Patil, Ms Varsha Galande, Mr Vedant Kekan, Ms Kalpana Dange: ‘*Sentiment Analysis Using Support Vector Machine’ International Journal of Innovative Research in Computer and Communication Engineering* (An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 1, January 2014.
7. Rodrigo Moraes, Joao Francisco Valiati, Wilson P. GaviaoNeto, ‘Document-level sentiment classification: An empirical comparison between SVM and ANN’, Expert Systems with Applications 40 621-633, 2013.
8. Thorsten Joachims: Text categorization with support vector machines: Learning with many relevant features, Proc. of ECML-98, 10th European Conference on Machine Learning, Springer Verlag, Heidelberg, DE, pp. 137-142, 1998.
9. Dr. S. Vijayarani1, Ms. J. Ilamathi2, Ms. Nithya3: ‘Preprocessing Techniques for Text Mining - An Overview’, *International Journal of Computer Science & Communication Networks*, Vol 5(1),7-16.
10. Cardie, C., Farina, C., Bruce, T., Wagner, E. 2006. Using natural language processing to improve eRulemaking. In Proceedings of Digital Government Research.
11. Kwon, N., Shulman, S., Hovy, E. 2006. Multidimensional text analysis for eRulemaking. In Proceedings of Digital Government Research. In Proceeding of the 2006 international conference on Digital government research, pp. 157-166.
12. Alessia D’Andrea, Fernando Ferri, Patrizia Grifoni, Tiziana Guzzo: ‘Approaches, Tools and Applications for Sentiment Analysis Implementation’, *International Journal of Computer Applications* (0975 – 8887) Volume 125 – No.3, September 2015.
13. Zhang, Bangzuo, and Wanli Zuo. "Reliable Negative Extracting Based on kNN for Learning from Positive and Unlabeled Examples", *Journal of Computers*, 2009.
14. Malutan, Raul, Pedro Gómez Vilda, and Monica Borda. "Combined Clustering Methods for Microarray Data Analysis", Advanced Engineering Forum, 2013.
15. https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization?lang=en
16. https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html
17. https://www.growthaccelerationpartners.com/blog/sentiment-analysis/
18. http://www.statsoft.com/Textbook/Support-Vector-Machines

How to cite this article:

Yash Pankhania *et al* (2018) 'Theoretical Study of Sentiment Analysis Using Support Vector Machine', *International Journal of Current Advanced Research,* 07(5), pp. 12843-12845. DOI: http://dx.doi.org/10.24327/ijcar.2018.12845.2272

****