**Sentiment Analysis in Python**

**Abstract**

Interest in sentiment analysis, particularly surrounding communication in Online Social Networks (OSNs), has grown rapidly in recent years. Python is one of the most popular programming languages for carrying out sentiment analysis, through the use of the Natural Language Toolkit (NLTK) and Scikit-learn. Studies on sentiment analysis mainly focus on the comparison between lexicon-based and machine learning based approaches, comparing popular machine learning methods, and the specific programming packages available. This paper presents an overview of sentiment analysis methods, a survey on the latest developments and usage, and an insight into the use of Python for such analysis. Finally, some possible future directions of research are pointed out.

**Introduction**

Sentiment analysis, or opinion mining, is the process of investigating, analysing and extracting emotion from subjective texts involving a person’s opinions, preferences and sentiment. Bo Pang put forth the concept in 2002, and the field has grown rapidly due to practicality in business, opinion monitoring and competitive intelligence in business (Pang, 2002). Generally speaking, there are two principal methods used to carry out sentiment analysis: a lexicon-based approach and a machine learning approach. Lexical-based methods make use of a predefined list of words, where each word is associated with a specific sentiment (Hu et al, 2013), while machine learning methods rely on supervised classification approaches, where sentiment detection is framed as a binary (i.e., positive or negative) (Goncalves, 2014). There are many different languages and programs available through which sentiment analysis can be carried out, but Python has recently emerged has one of the most powerful and most popular. Pedrogosa (2011) argues that due to Python’s high-level interactive nature and its maturing ecosystem of scientific libraries, it is an appealing choice for algorithmic development and exploratory data analysis. The purpose of this literature review is to discuss and analyse existing literature surrounding sentiment analysis and the use of python for such analysis.

**Use of Sentiment Analysis**

Sentiment analysis has been used in various applications, including analysis of the repercussions of events in social networks, analysis of opinions about a product or service, or to simply further understand social communication in Online Social Networks (OSNs) (Goncalves, 2014). Natural human language carries two types of information: objective information about facts and subjective information that contains opinions, emotions and personal sentiment (Nanli, Ping, Weiguo & Meng, 2012). Sentiment analysis focuses on deciphering and understanding the subjective components of language. Due to the rapid growth of blogs and review sites in recent times, this mass of subjective information is growing exponentially, which has brought increased power and popularity to sentiment analysis. Modern, real-world examples where it has been employed include: predicting volatility in the stock market (Bollen, Mao & Zeng, 2013), extracting indications of political sentiment from tweets (Tumasjan, Sprenger, Sandner, & Welpe, 2010), predicting the onset of depression in individuals based on text from social media (De Choudhury, Gamon, Counts, & Horvitz, 2013) and measuring national happiness based on Facebook status updates (Kramer, 2010). Sentiment analysis allows certain patterns to be understood, opinions to be gathered and collated, and predictions or decisions to be made.

**Approaches: Lexical vs Machine Learning**

The first approach to sentiment analysis to be discussed is the lexicon-based approach, which relies on an underlying sentiment (or opinion) lexicon. A sentiment lexicon is a list of lexical features (e.g., words) which are generally labelled according to their semantic orientation as either positive or negative (Liu, 2010). This measure of positivity or negativity is referred to as polarity. An example of such a lexicon that has been widely used in the social media domain is the Linguistic Inquiry and Word Count (LIWC) (Hutto & Gilbert, 2014). LIWC is text analysis software designed for studying the various emotional, cognitive and structural components present in text samples. LIWC uses a dictionary of almost 4,500 words organized into one (or more) of 76 categories. A second popular lexicon, The Affective Norms for English Words (ANEW), provides a set of normative emotional ratings for 1,034 English words (Bradley & Lang, 1999). Unlike LIWC, the words in ANEW have been ranked in terms of their pleasure, arousal, and dominance. Each word has a sentiment valence ranging from 1-9, with 5 being neutral. Words with valence scores less than five are negative, and those with scores greater than five are positive. However, neither of these dictionaries account for certain sentiment-bearing lexical items such as acronyms, initialisms, emoticons, or slang, which are known to be important for sentiment analysis of social text (Davidov, Tsur, & Rappoport, 2010). This is a major shortfall of the lexical approach to sentiment analysis.

The second set of approaches to sentiment analysis are those based on machine learning, of which Naïve Bayes, Maximum Entropy and SVM (Support Vector Machine) algorithms are most popular. Opinion on the effectiveness of machine learning methods in literature varies quite drastically. Goncalves (2014) argues that a primary advantage of machine learning methods is their ability to “adapt and create trained models for specific purposes and contexts”, but they are limited by the availability of labelled data, and labelling data may be “costly or even prohibitive for some tasks.” Some research concludes that machine learning approaches are more suitable for the analysis of OSN (Online Social Network) communication than the lexical-based LIWC method (Tausczik & Pennebaker, 2010). Cambria et al (2013) contradict this when they claim that these statistical methods are “semantically weak” and in fact “don’t work well on smaller text units such as sentences or clauses”.

Despite the wide use and popularity of both machine learning and lexical methods, it is unclear which are better for identifying the polarity (i.e., positivity or negativity) of a message (Goncalves, 2014). Manually creating and validating a comprehensive sentiment lexicon is labour and time intensive, and as a result much work has explored automated, or machine learning based, means of identifying sentiment relevant features in text (Hutto & Gilbert, 2014). Machine learning approaches, however, are not without their own drawbacks. The models require extensive training data which are, as with validated sentiment lexicons, often troublesome to acquire. The model also depends on the training set to represent as many features as possible, which is not always the case with short, sparse text of social media. Finally, they are often more computationally expensive in terms of CPU processing and training/classification time, which limits the models ability assess sentiment on streaming data such as a liveTwitter feed. (Hutto & Gilbert, 2014).

**Python in Sentiment Analysis**

The next body of literature reviewed focused on sentiment analysis in practice, particularly through the Python programming language. A widely used and perhaps the most popular python toolkit for sentiment analysis is the Natural Language Toolkit (NTLK), created by Edward Loper and Steven Bird. The NTLK is a suite of open source program modules, tutorials and problem sets that covers both symbolic and statistical natural language processing (Loper & Bird, 2002). It incorporates many corpora, such as the Stopwords Corpus or Brown Corpus, which are large collections of structured text. These corpora are then used for the pre-processing of data, specifically for part-of-speech (POS) tagging and tokenizing (Agarwal et al, 2016).

Stopword filtering is a central part of the sentiment analysis process. As mentioned above, the NLTK toolkit features a stopwords corpus, which is a list of frequently used words in eleven different languages that carry little lexical content. (Bird, Klein & Loper, 2009). These words are filtered out to increase the speed and efficiency of the analysis. As machine learning based methods can be computationally intense, literature concludes that the use of a stopword corpus, which saves on memory and reduces noise, is beneficial (Turdjai & Mutijarsa, 2016; Ganesan, 2018).

Literature appears to agree that the NLTK is suitable and efficient for the preparation of input for other sentiment analysis methods, such as SentiWordNet which was used by Agarwal, or scikit-learn, utilised by Samal (Agarwal et al, 2016; Samal et al, 2017). Samal et al (2017) conducted sentiment analysis, using supervised machine learning methods, on movie reviews. The NTLK was used for the processing of the data in python, while the scikit-learn Python package (Pedrogosa et al, 2011) was used for the training and testing of the model. Specifically, the NLTK was used to assign positive or negative labels to each one of the reviews in a large sample of movie reviews (Samal et al, 2017).

The literature also discusses a second toolkit available to carry out sentiment analysis in Python, developed by Pedrogosa et al, called Scikit-learn (Pedrogosa et al, 2011). Pedrogosa agreed with previous literature when they stated that “the Python programming language is establishing itself as one of the most popular languages for scientific computing”, and the reasons why this is the case are discussed previously. While the NLTK is primarily used for the pre-processing of data, scikit-learn offers implementation of many machine learning techniques and classifiers within Python. Scikit-learn has been used in practice to compare machine learning classifiers in predicting election results (Juneja & Ojha, 2017) and to develop a self-help sentiment analysis for medical treatments to be used by patients (Dandage et al, 2017). It is important to note that both of these studies used both the NLTK and Scikit-learn in the development, training and testing of their models. This illustrates that these packages are not in competition with each other, so to speak, but are complementary.

**Machine Learning Methods**

The final area of discussion in this paper is the comparison of the various machine learning classification methods for sentiment analysis in Python. The three most popular approaches are Naïve Bayes, Maximum Entropy and SVM (Support Vector Machine). Much of the literature focuses on all three methods, and comparing them across a number of scenarios and results (Hutto & Gilbert, 2014; Tausczik & Pennebaker, 2010; Juneja & Ojha, 2017).

Hutto & Gilbert (2014) describe the differences between the three approaches in a clear and concise manner. The Naive Bayes (NB) classifier is a simple classifier that relies on Bayesian probability and the naive assumption that feature probabilities are independent of one another. On the other hand, Maximum Entropy (ME) is a general purpose machine learning technique (an exponential model) using multinomial logistic regression. Unlike NB, ME makes no conditional independence assumption between features, and thereby accounts for information entropy, or lack of predictability (Hutto & Gilbert, 2014). Finally, Support Vector Machines (SVMs) differ from both NB and ME models in that SVMs are non-probability classifiers which operate by separating data points in space using one or more hyperplanes, and are thus more effective in higher dimensions (Scikit-learn.org, 2017).

Juneja and Ojha found the Naïve Bayes classifier to be most accurate of the methods tested, with an accuracy of 78% in classifying tweets directed at political parties as positive or negative (Juneja & Ojha, 2017). It is interesting to note that the accuracy of each of the methods depends on the nature of the dataset being analysed. Vaghela et al (2016) found, when analysing movie reviews, SVM (82.9%) performed better than ME (81%) and NB (81.5%). However, when analysing Twitter data, their results concurred with those of Juneja & Ojha. Using this data, NB had an accuracy of 88.2%, while ME achieved 83.8% and SVM managed 85.5% (Vaghela et al, 2016). It is evident that Naïve Bayes performs better on shorter text that may contain slang (Twitter data), while SVM is superior for longer, more formal text structures (movie reviews). It is interesting to note that Maximum Entropy classification was the least efficient in both cases.

**Conclusion**

It is clear that sentiment analysis, and particularly sentiment analysis using python, is a saturated and well researched field. There is much literature detailing and comparing lexical-based and machine learning based approaches to sentiment analysis. The literature discussed was found through the ABI/Inform Global and Scopus databases, with some package documentation and web articles considered when informative. There is a gap in the literature, however, in comparing the efficiency of different programming languages and different packages in programming languages for this analysis. A possible area for future study may be comparing the use of Python and C++ for sentiment analysis, and in turn comparing the NLTK and Scikit-learn packages within Python, on the basis of computational efficiency, ease of analysis, and accuracy.

**References**

Agarwal, A., Sharma, V., Sikka, G. & Dhir, R. (2016). *Opinion Mining of News Headlines Using Sentiwordnet*, Symposium on Colossal Data Analysis and Networking (CDAN).

Bird, S., Klein, E. & Loper, E. (2009). Natural language processing with Python. Beijing: O'Reilly.

Bollen, J., Mao, H., & Zeng, X.-J. (2010). *Twitter Mood Predicts the Stock Market*. CoRR, abs/1010.3003.

Bradley, M.M., & Lang, P.J. (1999*). Affective norms for English words (ANEW): Instruction manual and affective ratings*. Technical Report C-1. The Center for Research in Psychophysiology, University of Florida.

Cambria, E., Schuller, B., Xia, Y. & Havasi, C. (2013). *New avenues in opinion mining and sentiment analysis*, IEEE Intell. Syst. 28 (2) 15–21.

Davidov, D., Tsur, O., & Rappoport, A. (2010). *Enhanced Sentiment Learning Using Twitter Hashtags and Smileys*. ICCL-10.

De Choudhury, M., Gamon, M., Counts, S., & Horvitz, E. (2013). *Predicting Depression via Social Media*. In Proc. ICWSM-13.

Dondage, S. & Huber, J. (2018). *Patient Empowerment through Summarization of Discussion Threads on Treatments in a Patient Self-Help Forum.* IFMBE Proceedings, [online] 66, pp.229-233. Available at: http://dbis.eprints.uni-ulm.de/1573/1/ICBHI\_paper35\_masterfile.pdf [Accessed 3 Apr. 2018].

Ganesan, K. (2018). *Why Use Stop Words for Text Mining? | RxNLP*. [online] Machine Learning & Text Mining Tools and Tutorials for Language Processing at Scale. Available at: http://rxnlp.com/why-use-stop-words-for-text-mining/#.WsOayS7wbIU [Accessed 3 Apr. 2018].

Hu, X., Tang, J., Gao, H. & Liu H. (2013). *Unsupervised sentiment analysis with emotional signals*. In International Conference on World Wide Web (WWW)

Juneja,, P. & Ojha, U. (2017). *Casting online votes: To predict offline results using sentiment analysis by machine learning classifiers*, 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, pp. 1-6.

Kramer, A. (2010). *An unobtrusive behavioral model of “gross national happiness.”* In Proc. CHI-10.

Liu, B. (2010). *Sentiment Analysis and Subjectivity*. In N. Indurkhya & F. Damerau (Eds.), Handbook of Natural Language Processing (2nd ed.). Boca Raton, FL: Chapman & Hall

Loper, E. & Bird, S. (2002) *NLTK: The Natural Language Toolkit* [online] Available at: <https://arxiv.org/pdf/cs/0205028.pdf> [Accessed 4 Apr. 2018].

Pang, B., Lee, L. & Vaithyanathan, S. (2002). *Thumbs up?: sentiment classification using machine learning techniques*, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10, pp. 79– 86.

Pedregosa et al. (2011). *Scikit-learn: Machine Learning in Python*, JMLR, vol. 12, pp. 2825-2830.

Samal, B.R., Behera, A.K. & Panda, M. (2017). *Performance analysis of supervised machine learning techniques for sentiment analysis*, Proceedings of 2017 3rd IEEE International Conference on Sensing, Signal Processing and Security, ICSSS 2017, pp. 128.

Scikit-learn.org. (2017). *1.4. Support Vector Machines — scikit-learn 0.19.1 documentation*. [Online] Available at: http://scikit-learn.org/stable/modules/svm.html [Accessed 3 Apr. 2018].

Tausczik Y. R. & Pennebaker J. W. (2010). *The psychological meaning of words: LIWC and computerized text analysis methods*, Journal of Language and Social Psychology, 29(1):24–54.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). *Predicting Elections with Twitter*, In Proc. ICWSM-10.

Turdjai A. A. & Mutijarsa, K. (2016). *Simulation of marketplace customer satisfaction analysis based on machine learning algorithms*, 2016 International Seminar on Application for Technology of Information and Communication (ISemantic), Semarang, pp. 157-162.

Vaghela, V. B., Jadav, B. M. & Scholar, M. E. (2016). *Analysis of Various Sentiment Classification Techniques*, Int. J. Comput. Appl., vol. 140, no. 3, pp. 975–8887.