C0MP90049 Knowledge Technologies Project 2
Sentiment Analysis

Problem Description

Social media platforms such as Twitter captures the opinions of their users through microblogs (known as tweets) posted by their users on their websites. The sentiment of these tweets can be manually identified as positive, negative, or neutral, and can then be used to train machine learning models in order to predict the sentiment of other unlabelled tweets. The use of raw text data, however, presents a set of problems for the training process. Should the word "happy" and "happiness" be classed as a separate feature to the training process when they both convey the same sentiment? Should frequently occurring words such as "the", "these", and "are" be used as a feature when they convey no sentiment at all? Should the number of times that a feature appears in a tweet be used to quantify the magnitude of the sentiment that it is trying to convey? And ultimately, is it possible to use tweet text to help us to identify people sentiment on Twitter?

In this report, we will examine how varies methods of preprocessing raw text data can be used to effectively train a machine learning model to increase its prediction performance in predicting the sentiment of tweets text.

Literature Review

In this section, we will examine the varies preprocessing techniques and machine learning models that have been used in the literature to analyse the sentiment of short text.

Lemmatisation

The word "happily" and "happiness" are both derived from the root word "happy", and as such, should be considered as the same word (base form) as they both convey the same sentiment.

There are two widely discussed methods for reducing words into its base form, Stemming, and Lemmatisation. Asghar et al. have compared both methods and concluded that lemmatisation is considered to be the more accurate method. In their article, they have mentioned that "the words "caring" and "cars" are reduced to "car" in a stemming process, whereas lemmatisation was able to reduce it to "care" and "car" respectively" (Asghar et al. 2014).

Removal of Stopwords

Saif et al. define stopwords as "meaningless words that have low discrimination power" in a document. Stopwords are words such as "the", "of", or "an", and they occur frequently in a document but do not contribute to the sentiment of the document. Removing these words will optimise the training process of the machine learning model and reduce the size of the features list. This will, in turn, reduce the complexity of the trained model, without losing crucial information (Krouska, Troussas, & Virvou 2016; Saif et al. 2014; Asghar et al. 2014).

Feature weighting - TF-IDF

It seems intuitive to characterise a tweet by the number of times at which a feature appears in it, but research has shown that this process actually reduces the accuracy of the sentiment analysis process (Mccallum and Nigam, 1998; Pang et al., 2002). To offset this problem, the feature count is transformed into the TF-IDF value as follows:

$$\text{TF-IDF} = \text{TF} \cdot \text{IDF}$$

$$\text{TF} = \frac{\text{\# features in the document}}{\text{\# of terms in the document}}$$

$$\text{IDF} = \frac{\text{\# of documents in the collection}}{\text{\# of documents containing the term}}$$

The above computation of TF-IDF weights the term-frequency (TF) is weighted by the inverse-document-frequency (IDF), where IDF is a number that quantifies how rare a feature is out of the entire document collection. As such, "the utilisation of IDF in information retrieval is based on its ability to distinguish between content-bearing words (words with some semantical meaning) and simple function words" (Paltoglou & Thelwall 2010).

Support Vector Machine

There are many articles in the literature that have praised the Support Vector Machine (SVM) as the best machine learning model for sentiment analysis (Paltoglou & Thelwall 2010; Basari et al. 2013; Ohana & Tierney 2009; Haddi, Liu & Shi 2013). Most notably, Pang et al. (2002) has compared the performance of three machine learning models, namely Naive Bayes, Maximum Entropy Classification, and SVM on a data with a diverse set of features and concluded that SVM performed with an average accuracy of more than 80%. As such, the SVM will be the machine learning model of choice for this experiment.

Methodology

Dataset

The dataset for this experiment consists of 27913 tweets, each having its sentiment manually labeled as positive, negative, or neutral (Rosenthal, Noura, & Nakov 2017). The entire dataset is divided into a set for training the machine learning model (train-tweets.txt) and a set for evaluating the performance of the trained model (eval-tweets.txt).

Hypotheses

We will prove the following hypotheses in this experiment :

1. **Proper preprocessing of raw tweet data will improve the performance of the Support Vector Machine when used to predict the sentiment of tweets**

2. **Weighting feature count with TF-IDF will further improve the performance of Support Vector Machine when used to predict the sentiment of tweets**

Evaluation Methods

Three sets of data will be used to compare the performance of its respectively trained SVM. The experiment will be implemented on python, and utilises the Scikit-learn and the spaCy library.

<u>"Badly Vectorised" Dataset</u>
Provided for this experiment are the files train.csv and eval.csv containing an incomplete vectorisation of the collection of tweets. Only 45 features were captured in the vectors, and stopwords are not removed.

<u>"Preprocesed" Dataset</u>
A new set of data were then generated from the collection of tweets to capture most of its relevant features using the following processes:

1. Tweets were tokenised, with unnecessary white spaces and punctuations removed. This was done using the inbuilt tokeniser from the spaCy library.

2. Stopwords were removed. The list of stopwords were taken from the the spaCy library consisting of 326 stopwords.

3. Each token was transformed into its base form via a lemmatisation algorithm provided in the spaCy library.

4. The processed tweets were then vectorised on its feature count, with the resultant vectors capturing 48225 features from a collection of tweet 27913. This was done using `CountVectorizer()` from the scikit-learn library.

<u>"TF-IDF Weighted" Dataset</u>
The feature counts from the "pre-processed" dataset were then transformed into its respective TF-IDF value

to procured this dataset. This was done using the `TfidfVectorizer()` from the scikit-learn library.

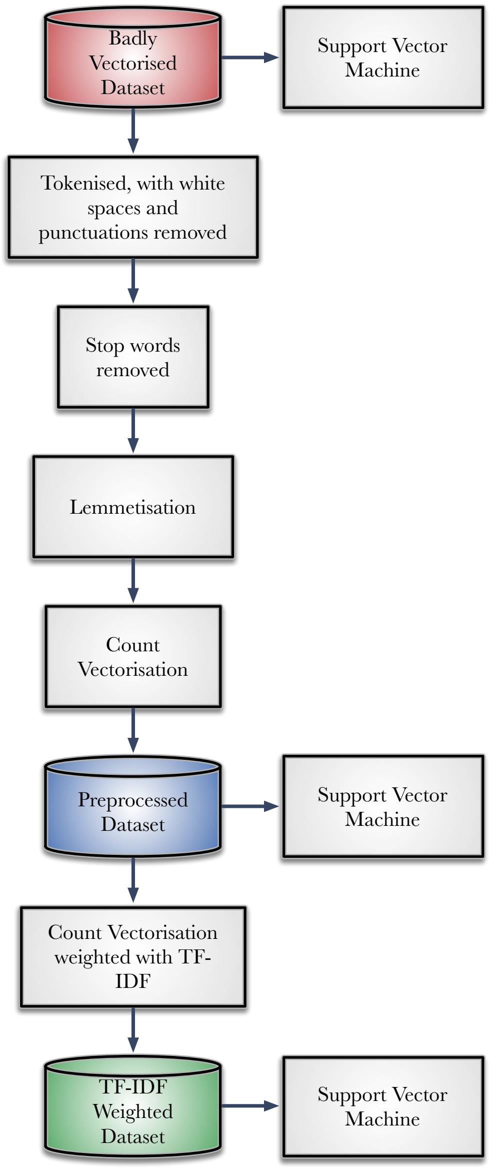

Figure 1: Flowchart of experiment's processes

Evaluation Metrics

Because both precision and recall are of equal importance in quantifying the performance of the SVM's prediction, the harmonic mean of the precision and recall known as the F-measure will be used instead (Hripcsak & Rothschild 2005; Haddi, Liu & Shi 2013). In this experiment, the F-measure and the accuracy will be used to quantify the performance of the model.

Results

Figure 2: F-Measure of SVM

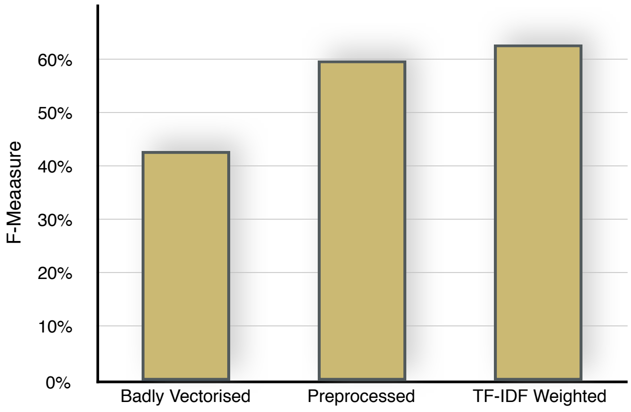

Figure 3: Accuracy of SVM

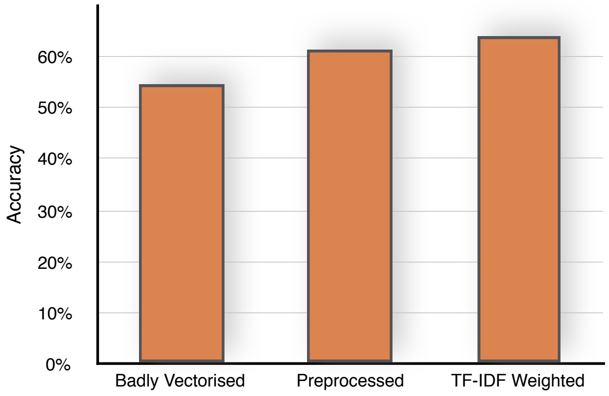

Table 1: F-Measure & Accuracy of SVM

	Badly Vectorised	**Preprocessed**	**TF-IDF Weighted**
F-Measure	42.56%	59.50%	62.48%
Accuracy	54.67%	61.04%	63.84%

Table 1 shows the f-measured and accuracy of the performance of the model using the three datasets as training data. From figure 1 and 2, a general increase in F-measure and accuracy can be observed. The results therefore reveal that proper preprocessing and vectorisation of tweets have led to an increase in the performance of the SVM in predicting the sentiment of tweets. Transforming feature counts into its TF-IDF value also further increased the prediction performance. This is an **overall improvement of 46.8%** in F-measure and **16.78%** in accuracy. The improvements in prediction performance are significant, and as such, both hypotheses mentioned earlier in this report can be accepted.

Discussion & Conclusion

This experiment has shown that using appropriate preprocessing and transforming feature counts to the TF-IDF value has resulted in an improvement to performance in sentiment prediction of tweets. As such, this experiment has shown that we, in fact, can use tweet text to help us to identify people's sentiment on Twitter.

The performance of the sentiment analysis, however, can be further improved by applying additional preprocessing methods. Research has shown that improvement in performance can be attained by dealing with negation of words ("not happy" is not to be treated the same as "happy"), emoticons, and transforming abbreviations back to its full words (Angiani et al. 2016; Haddi, Liu & Shi 2013; Mohammad, Kiritchenko & Zhu 2013).

Bibilography

Angiani, G., Ferrari, L., Fontanini, T., Fornacciari, P., Iotti, E., Magliani, F. and Manicardi, S., 2016. A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter. In KDWeb.

Asghar, M.Z., Khan, A., Ahmad, S. and Kundi, F.M., 2014. A review of feature extraction in sentiment analysis. Journal of Basic and Applied Scientific Research, 4(3), pp.181-186.

Basari, A.S.H., Hussin, B., Ananta, I.G.P. and Zeniarja, J., 2013. Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization. Procedia Engineering, 53, pp.453-462.

Pang, B., Lee, L. and Vaithyanathan, S., 2002, July. 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). Association for Computational Linguistics.

Haddi, E., Liu, X. and Shi, Y., 2013. The role of text pre-processing in sentiment analysis. Procedia Computer Science, 17, pp.26-32.

Hripcsak, G. and Rothschild, A.S., 2005. Agreement, the f-measure, and reliability in information retrieval. Journal of the American Medical Informatics Association, 12(3), pp.296-298.

Krouska, A., Troussas, C. and Virvou, M., 2016, July. The effect of preprocessing techniques on Twitter sentiment analysis. In 2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA) (pp. 1-5). IEEE.

McCallum, A. and Nigam, K., 1998, July. A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization (Vol. 752, No. 1, pp. 41-48).

Mohammad, S.M., Kiritchenko, S. and Zhu, X., 2013. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. arXiv preprint arXiv:1308.6242.

Ohana, B. and Tierney, B., 2009. Sentiment classification of reviews using SentiWordNet.

Paltoglou, G. and Thelwall, M., 2010, July. A study of information retrieval weighting schemes for sentiment analysis. In Proceedings of the 48th annual meeting of the association for computational linguistics (pp. 1386-1395). Association for Computational Linguistics.

Rosenthal, Sara, Noura Farra, and Preslav Nakov (2017). SemEval-2017 Task 4: Sentiment Analysis in Twitter. In Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval '17). Vancouver, Canada.

Saif, H., Fernández, M., He, Y. and Alani, H., 2014. On stopwords, filtering and data sparsity for sentiment analysis of twitter.