COMP90049 Introduction to Machine Learning
Project 3 Report

1 Introduction

Twitter is one of the most popular microblogging and social network services which allows users to post and interact with messages called “tweets”. People can post tweets to discuss current issues, share opinions, express emotions or share opinions. The great amount of shared data on Twitter can be valuable for learning users sentiment and opinions about a variety of topics. This creates a challenge for automatically extracting sentiments from tweets. In this project, different machine learning models are investigated to predict tweet sentiment. The validity of each model is assessed by using the accuracy score.

The rest of the paper is organised as follows. Section 2 is a brief summary of four related literatures. Section 3 explains the feature engineering techniques, machine learning models and evaluation metrics used in the project. The performance results of each model are presented in Section 4. Section 5 provides detailed analysis about the behaviours of each model and some limitations. Section 6 concludes the paper and discusses some future directions.

2 Literature review

Many previous studies have been conducted on sentiment analysis. Bermingham and Smeaton (2010) compare the difficulty between classifying sentiment in short documents and long documents. They report that it is easier to classify microblogs and most machine learning classifiers can achieve accuracy of greater than 70%. This result offers a compelling reason for conducting microblog sentiment analysis.

There is a great number of research papers that discuss sentiment classification on tweet datasets. Agarwal (2011) et al. classify sentiment by using tree kernel and feature based models and prove that both models achieve better performance than the unigram baseline. Go, Bhayani and Huang (2009) found that using emoticons is an effective way to classify training data. They conclude that the machine learning models, including Naive Bayes, Maximum Entropy and Support Vector Machines that are trained with emoticon data can achieve accuracy of above 80%. In addition, Vadicamo et al. (2017) investigate approaches to predict sentiment of tweets with both text and images. They show that text associated with images can help to train a Convolutional Neural Network model that effectively classifies the sentiment.

3 Methods

Holdout method is used in this project. The Tweets dataset is divided into three parts: a labelled training set, a labelled development set that is used for selecting model and tuning, and an unlabeled test set that is used for evaluating performance. In the training and

development set, each tweet is labelled as pos, neg or neu, indicating positive, negative and neutral sentiments separately. Here is an example tweet in the training dataset.

sentiment	tweet_id	tweet
neg	1	@bullyosullivan oh no! so sorry about your pets..

Since all the raw tweets are strings, some feature engineering techniques have been applied to raw tweet dataset with the purpose of improving the performance of the machine learning models. The feature engineering techniques include Bag of Words (BoW), Term Frequency-Inverse Documentation Frequency (TF-IDF) and Global Vectors for Word Representation (GloVe). The machine learning models are K Nearest Neighbour, Naive Bayes, Logistic Regression and Multilayer Perceptron. Besides, accuracy score is used as the evaluation metrics to evaluate the validity of each model. In addition, the models and feature engineering techniques are treated as two distinct components, which makes it easier to test the performance of different combinations of models and feature engineering techniques.

3.1 Feature Engineering Techniques

3.1.1 Bag-of-Word (BoW)

After filtering out highly frequent and infrequent words, each word is matched to a unique ID. Hence, in BoW, each tweet is represented as a list of tuples [(word_id, word_count)]. Word_count is the number of occurrences of a word in a tweet.

3.1.2 Term Frequency-Inverse Documentation Frequency (TF-IDF)

TF-IDF is the same as BoW, except that it represents each word by its TF-IDF value rather than the number of occurrences.

3.1.2 Global Vectors for Word Representation (GloVe)

In GloVe, each tweet is represented as a single 100-dimensional vector. The vector is obtained by summing the 100-dimensional GloVe embedding vector of each word in the tweet.

3.2 Machine Learning Models

3.2.1 Baseline

The Zero R, also known as majority class, is chosen as a baseline. In this project, all the other machine learning models are compared against the Zero R. It is the most commonly used baseline in machine learning. The implementation of Zero R is also simple, which only requires classifying all instances according to the most frequent label in the training set.

3.2.2 K Nearest Neighbour (KNN)

K Nearest Neighbour is a simple and easy-to-implement machine learning model that can be used for classification problems. It is a lazy approach that simply stores all training data and returns the most common class labels among K nearest neighbours for each testing instance (Han, Kamber & Mining, 2006). In the project, K is set to 101 which is an optimal value. The number is odd to avoid the tie breaking issue.

3.2.3 Naive Bayes

Naive Bayes is a probabilistic generative model that is built based on Bayes' theorem (Eisenstein, 2019). It applies the method of maximizing likelihood and makes prediction by maximizing the joint probability $P(x, y)$.

$$P(x, y) = P(y)P(x|y) = \prod_{i=1}^{N} P(y^i) \prod_{m=1}^{M} P(x_m^i|y^i)$$

In this project, BernoulliNB and Multinomial Naive Bayes are implemented since they are one of the two classic Naive Bayes algorithms that can perform well on text classification tasks.

3.2.4 Logistic Regression

Logistic Regression is a probabilistic discriminative model. Unlike Naive Bayes, it finds the optimal parameters by directly maximizing $P(y|x)$ (Shalizi, 2013).

$$P(x, y) = P(y)P(x|y) = P(y|x)P(x)$$
$$\hat{y} = argmax\, P(y|x)$$

In this project, since there are three categories of tweet sentiment, Multinomial Logistic Regression is used.

3.2.5 Multilayer Perceptron

The Multilayer Perceptron is a neural network that consists of at least three layers: an input layer, a hidden layer and an output layer (Svensén & Bishop, 2007). This project uses two hidden layers that are set to 10 and 5. This is because too complicated models can easily lead to the overfitting problem. The parameter early_stopping is set to true, which also aims to prevent overfitting. Besides, the sigmoid function is applied as the activation function.

3.3 Evaluation Metrics

3.3.1 Accuracy Score

Accuracy score is a basic evaluation metric that can be used to assess the performance of each machine learning model. Accuracy score is calculated by applying the following formula.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ test\ instances}$$

4 Result

Method/Accuracy	BoW	TF-IDF	GloVe
Zero R	40.67%	40.67%	40.67%
KNN (K = 101)	62.44%	69.76%	62.81%
Multinomial Naive Bayes	73.08%	72.56%	44.02%
Bernoulli Naive Bayes	73.38%	73.38%	58.89%
Logistic Regression	74.09%	74.17%	67.79%
Multilayer Perceptron (hidden_layer = (10, 5))	74.41%	74.36%	68.63%

Table 1 - Models Accuracy Scores

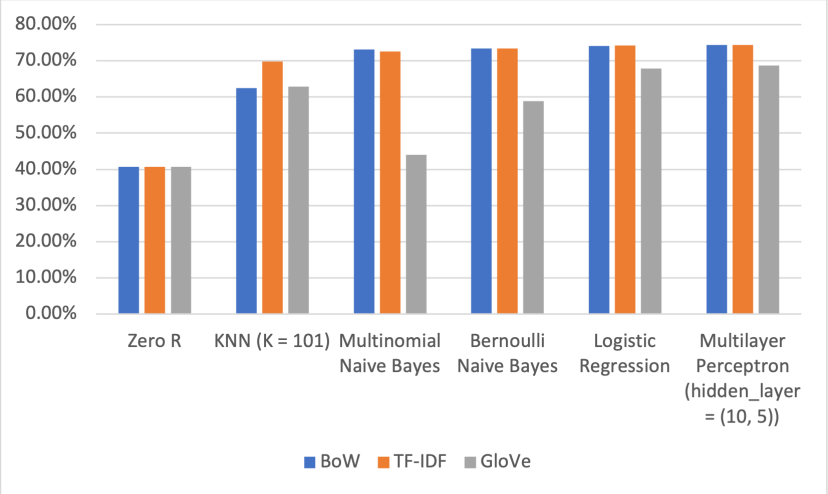

Figure 1 - Models Accuray Score Bar Chart

Method/Accuracy	BoW	TF-IDF	GloVe
Multilayer Perceptron (hidden_layer = (10, 5))	74.41%	74.36%	68.63%
Multilayer Perceptron (hidden_layer = (50, 25))	74.32%	74.28%	70.33%
Multilayer Perceptron (hidden_layer = (100, 50))	74.17%	73.92%	70.85%

Table 2 - Multilayer Perceptron Models Accuracy Scores

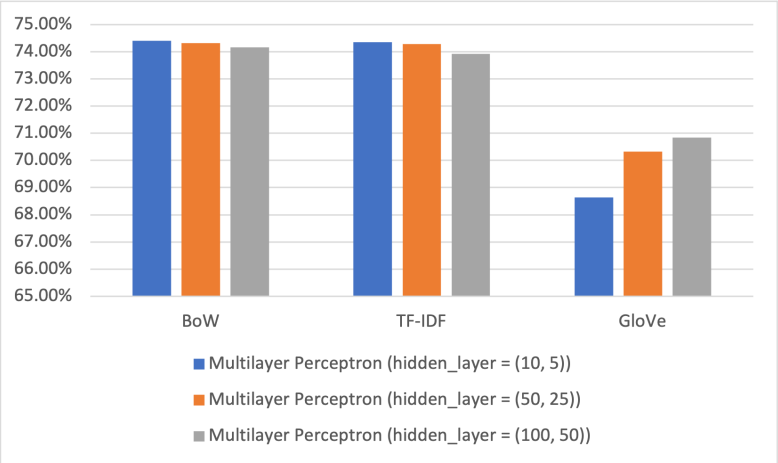

Figure 2 - Multilayer Perceptron Models Accuracy Scores Bar Chart

5 Critical Analysis

5.1 Contextualises Models Behaviours

Figure 1 displays the accuracy obtained by all machine learning models in the development dataset. The result obtained by Zero R is also shown as a baseline reference. It is clear that all the machine models outperforms the baseline.

5.1.1 BoW vs. TF-IDF

According to Figure 1, in most machine learning models, TF-IDF performs better than BoW. The reason behind is that BoW representation only contains the number of occurrences of words. However, TF-IDF representation captures the importance of the words as well. In TF-IDF, words that are commonly present in all tweets are assigned a lower weight and therefore are less important. By contrast, words that are present many times in a few tweets are assigned a higher weight and therefore more valuable for making predictions (Sch¨utze, Manning & Raghavan, 2008).

5.1.2 KNN vs. Naive Bayes

As Figure 1 shows, with respect to BoW and TF-IDF datasets, Naive Bayes has higher accuracy scores. With respect to the GloVe dataset, however, KNN with K = 101 achieves higher accuracy. Compared to Naive Bayes, KNN is simpler to implement and does not rely on an unrealistic assumption. However, KNN has a drawback that is the cost incurred by

calculating the distance between a new instance and all existing instances can be huge for a large dataset. This may make KNN slower than Naive Bayes.

5.1.3 Logistic Regression vs. Naive Bayes

Generally, Logistic Regression outperforms Naive Bayes on large datasets. This is corresponding to the results in Figure 1 which shows that Logistic Regression has higher accuracy scores in all datasets. Naive Bayes makes a simplifying assumption that features are independent of each other. However, Logistic Regression does not rely on this assumption. Thus, if a dataset contains many correlated features, Logistic Regression will produce more accurate prediction than Naive Bayes. In the tweet dataset, it is obvious that some words in a tweet are correlated with each other. Hence, Logistic Regression achieves better performance than Naive Bayes.

5.1.4 Multilayer Perceptron vs. Naive Bayes & Logistic Regression

Among all machine learning models, Multilayer Perceptron achieves the highest accuracy score. Compared to Naive Bayes and Logistic Regression, Multilayer Perceptron has almost similar accuracy on BoW and TF-IDF datasets, but achieves higher accuracy on GloVe datasets. Unlike Bag of Word and TF-IDF which represent each word in a tuple, GloVe captures the meaning of each word and represents it as a 100-dimensional vector. When the number of features is large, a deep learning model, such as Multilayer Perceptron with greater learning capability is more applicable.

Despite the fact that Multilayer Perceptron is more powerful than Naive Bayes and Logistic Regression, it has a higher risk of overfitting. As Figure 2 shows, increasing the number of hidden layers, or nodes in the hidden layer to a large number sometimes decreases the performance of Multilayer Perceptron. This is caused by the overfitting issue. Thus, the parameter tuning process should be applied to find the optimal number of hidden layers.

5.2 Ethical Issues

Although all machine learning models achieve higher performance than the baseline, they still have some limitations. The main focus of this project is English Tweets. All the machine learning models will only work on tweets written in English, since those that are not written in English are discarded from the training datasets. However, Twitter users come from a wide range of countries, the machine learning models should be extended to handle Twitter written in other languages.

6 Conclusion

This project investigates the task of classifying tweet sentiment into positive, neutral and negative classes. Multiple Feature engineering techniques (Bag of Word, TF-IDF and GloVe) have been applied to raw tweet data to improve model performance. Various machine learning models (K Nearest Neighbour, Naive Bayes, Logistic Regression, Multilayer Perceptron) have been experimented on different datasets. Models performance have been

evaluated by using the accuracy score. The results demonstrate that all the machine learning models outperform the Zero R baseline and Multilayer Perceptron achieves the highest performance among all models. In future work, it can be interesting to explore more machine learning models or build a hybrid model by combining two or more different models.

7 Reference

Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. J. (2011, June). Sentiment analysis of twitter data. In *Proceedings of the workshop on language in social media (LSM 2011)* (pp. 30-38).

Bermingham, A., & Smeaton, A. F. (2010, October). Classifying sentiment in microblogs: is brevity an advantage?. In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 1833-1836).

Manning, C., & Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Eisenstein, J. (2019). *Introduction to natural language processing*. MIT press.

Han, J., Kamber, M., & Mining, D. (2006). Concepts and techniques. *Morgan Kaufmann*, *340*, 94104-3205.

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, *1*(12), 2009.

Vadicamo, L., Carrara, F., Cimino, A., Cresci, S., Dell'Orletta, F., Falchi, F., & Tesconi, M. (2017). Cross-media learning for image sentiment analysis in the wild. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (pp. 308-317).

Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval* (Vol. 39, pp. 234-265). Cambridge: Cambridge University Press.

Shalizi, C. (2013). Advanced data analysis from an elementary point of view.

Svensén, M., & Bishop, C. M. (2007). Pattern recognition and machine learning.

Díaz, M., Johnson, I., Lazar, A., Piper, A. M., & Gergle, D. (2018, April). Addressing age-related bias in sentiment analysis. In *Proceedings of the 2018 chi conference on human factors in computing systems* (pp. 1-14).

Yang, Y., & Eisenstein, J. (2017). Overcoming language variation in sentiment analysis with social attention. *Transactions of the Association for Computational Linguistics*, *5*, 295-307.