Sentiment Classification of Tweets

1 Introduction

Twitter is a microblogging and social networking platform where users post "tweets" which are their opinions and feelings. Sentiment of tweets can be classified into "positive", "neutral" and "negative". Twitter sentiment classification could potentially be used to gather feedback from authors about particular topics. This report aims is evaluate the behavior of three representations of raw tweets dataset published in (Vadicamo et al., 2017) and (Go et al., 2009) in various supervised machine learning models.

In this report, performance of feature "count", "tfidf" and "glove100" in logistic regression model (LR), k-nearest neighbors model (KNN), decision tree model (DT) and multi-layer perceptron model (MLP) will be analyzed to conclude the combination of feature and model with the greatest predict ability.

2 Literature review

There are already many papers written on sentiment analysis for different kinds of information. (Go et al., 2009) explored that some words are not useful for sentiment classification, they may express different meanings in different scenarios. Work from (Vadicamo et al., 2017) takes sentiment classification beyond Natural Language Processing (NLP) area to image-based sentiment analysis. Their result suggests that even though associated textual information could be unclear and noisy, it is still helpful in training visual sentiment classifier. Addition to plain textual information, (Agarwal et al., 2011) proposed that "parts-of-speech tags" of tweets such as emoticons and hashtags should marginally be added into classifiers. Instead of adding more features, (Kumar et al., 2012) focused on analyzing part of speech of opinion words in tweets. They proposed that opinion words are combinations of adjectives, verbs, and adverbs. Different methods could be used on identifying the semantic orientation of different parts of speech to increase accuracy. Apart from selecting features and training sentiment classifiers, (Saif et al., 2012) introduced a novel method of treating semantics of entities in tweets as an additional feature. Semantic of each entity can also represent sentiments to a certain extent, and this additional feature has increased the accuracy score of classifying negative and positive sentiment by 6.5% and 4.8% respectively.

3 Method

3.1 Evaluation Metrics

In this report, four evaluation metrics are used to evaluate the performance of multi-label supervised machine learning models (El Kafrawy et al., 2015). Considering multi-class evaluation, the macro-averaging approach is used to calculate precision and recall, since it is more affected by minority classes comparing to other approaches. Since we are equally interested in all three classes, we will focus more on the accuracy score and F1 score for the general performance of models.

a) Accuracy: Percentage of correctly predicted labels among all predicted and true labels.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

b) Precision: Percentage of correctly predicted labels among all true-predicted labels.

$$Precision = \frac{TP}{TP + FP}$$

c) Recall: Percentage of correctly predicted labels among all true labels.

$$Recall = \frac{TP}{TP + FN}$$

d) F1 Score: Harmonic mean between precision and recall.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

True Positive (TP): model correctly predicts the positive class

True Negative (TN): model correctly predicts the negative class

False Positive (FP): model incorrectly predicts the positive class

False Negative (FN): model incorrectly predict the negative class

3.2 Feature selection

Three features are selected to train different machine learning models in order to evalu-

ate their performance.

a) count: Words in the data set are filtered to remove those with extremely high and extremely low occurrence rates. The remaining words are mapped to a unique ID, counts of word occurrence are mapped to each ID.

b) tfidf: Words in the data set are filtered like the filtering process in the count feature. The tfidf values are calculated and mapped to each unique word ID.

$$TF = \frac{count\ of\ certain\ word\ in\ tweet}{number\ of\ word\ in\ tweet}$$

$$IDF = \log\left(\frac{number\ of\ tweets}{no.of\ tweets\ with\ certain\ word + 1}\right)$$

$$TF - IDF = TF * IDF$$

c) glove100: Each word is mapped into a 100-dimensional Glove vector which contains the meaning of the word. Vectors of words in each tweet will be summed to a single 100-dimensional vector.

3.2 Baseline

Weighted Random algorithm is implemented as the baseline of this report by weighting each class according to class prior probabilities, then randomly assigning a class to each instance.

Accuracy	0.38
Precision	0.34
Recall	0.34
F1 Score	0.33

Table 1: Weighted Random Baseline

3.3 Logistic Regression

The logistic regression model is a simple regression model, and it has no restrictive assumptions on features. The reason for using LR to classify sentiments of tweets is that LR is suitable for frequency-based features. LR is expected to work well with count features and tfidf features.

The probabilities of classes are represented as functions of feature x and parameters θ. In multiclass classification, the probabilities of instances falling into each class are computed by passing a generalization of the sigmoid function called the softmax function. A parameter vector θ_c is learnt for each class c.

$$p(y = c|x;\theta) = \frac{\exp(\theta_c)}{\sum_k \exp(\theta_k x)}$$

"lbfgs" is chosen to be the solver of LR since it is a multi-class solver. Among all multi-class solvers "lbfgs" has a fast linear convergence rate on large scale datasets. (Moritz et al., 2016)

3.4 K-nearest Neighbors

K-nearest Neighbors can be used as an interpretable classification model. The training phase is relatively simple compared to other methods. In the training phase, we only need to store all the training instances. In the testing phase, distances of a test instance to all training instances needs to be computed, then find K closest training data points and analysis the label of test instance based on labels of K closest train instances.

It is vital to choose a suitable K value. If the K value is rather large, the model will be simple and ignore plentiful information, if the K value is rather small, the model will have a high chance to be overfitting.

Following holdout evaluation strategy, a reasonable range of K value were tested on each feature. K value of 23 outperformed other K values by comparing their evaluation metrics.

3.5 Decision Tree

Decision Tree is another simple interpretable classification model. In a decision tree classification model, class labels are allocated at tree leaves, the branches that lead towards tree leaves represent conjunctions of attributes to partition the node instances.

The goal of constructing high-quality small trees points to the need for finding the suitable feature to be branches. The purity of classes needs to be calculated to ensure each partition is as homogeneous as possible. Entropy (H) is calculated as a measure of unpredictability, furthermore, information gain (IG) measures the reduction of entropy before and after the partition of data using a certain feature. To avoid the model becoming overfitting. Gain Ratio (GR) is introduced to punish features that lead to overfitting.

$$H = -\sum_i^n P(i)log_2P(i)$$

$$IG = H - \sum_i^m weight(x_i) * H(x_i)$$

$$GR = \frac{IG}{-\sum_i^m weight(x_i)log_2 weight(x_i)}$$

Another critical parameter of DT is the

maximum depth of the tree. The tree needs to stop expanding at a suitable stage to avoid becoming overfitting. A reasonable range of maximum depth of the tree was tested by holdout strategy, a suitable maximum depth of the tree is decided to be 9.

3.6 Multi-layer Perceptron

Multi-layer perceptron consists of three layers: input layer, hidden layers, and output layers. Each layer is fully connected to the neighboring layers. Each neuron within hidden layers uses a non-linear activation function to produce input for the next layer. The algorithm learns features as intermediate representations and produces a probability distribution over classes in multiclass classification. Multi-layer perceptron generally performs well because of its automatic feature learning property. With the limitation of GPU power, the potential of MLP might not be fully explored.

The hidden layer is designed to have two layers with 2 neurons in the first layer and 2 neurons in the second layer. Due to the limitation of GPU power, large-scale tests can not be used to determine the hidden layer sizes. The number of hidden layers and the number of neurons in each layer is decided based on experience and a small number of tests.

The solver parameter is selected to be "adam" which is a stochastic gradient-based optimizer. Solver "adam" has lesser training time and greater accuracy on relatively large datasets

The activation parameter is selected to be hyperbolic tan function since tan function obtains better recognition accuracy than other functions in the majority of MLP applications (Karlik et al., 2011).

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

4 Results

4.1 Logistic Regression Result

	count	tfidf	glove100
Accuracy	0.74	0.74	0.68
Precision	0.75	0.76	0.69
Recall	0.76	0.76	0.7
F1 Score	0.76	0.76	0.7

Table 2: Logistic Regression Model Result

4.2 K-nearest Neighbors Result

	count	tfidf	glove100
Accuracy	0.64	0.66	0.64
Precision	0.66	0.69	0.7
Recall	0.64	0.66	0.63
F1 Score	0.65	0.67	0.65

Table 3: K-nearest Neighbors Model Result

4.3 Decision Tree Result

	count	tfidf	glove100
Accuracy	0.46	0.46	0.59
Precision	0.57	0.57	0.62
Recall	0.4	0.4	0.6
F1 Score	0.35	0.35	0.61

Table 4: Decision Tree Model Result

4.2 Multi-layer Perceptron Result

	count	tfidf	glove100
Accuracy	0.74	0.72	0.68
Precision	0.75	0.75	0.7
Recall	0.75	0.73	0.7
F1 Score	0.75	0.74	0.7

Table 5: Multi-layer Perceptron Model Result

5 Discussion

5.1 System Behavior

In the LR result, we can observe that the performance of the count feature and tfidf feature are extremely close, and they both outperformed the glove100 feature. LR model has higher suitability to frequency-based features, thus frequency-based features like count and tfidf will obtain higher accuracy than word-embedding features like glove100. Count feature focuses more on the occurrence times of words while the tfidf feature focuses on the frequency of words appearing in tweets. Word with a high tfidf value generally has higher importance in the text which means it could represent the text better. Thus, the tfidf feature performs the best in the LR model.

In the KNN result, the tfidf feature performs slightly better than the other two features while the glove 100 feature has a greater precision score comparing to the count feature. In the KNN model, labels of K nearest nodes will be used to generate predictions, it is reasonable that tweets with the same high-frequency words are more likely to express the same sentiment. The difference

in precision score between the count feature and the glove100 feature is because glove100 feature learns by constructing a co-occurrence matrix that counts the frequency of words appearing in the whole context. Glove100 could capture relationships between word combinations rather than words. Thus, the predictions have a higher chance to be correct once the model makes the true prediction.

In the DT result, all three features performed poorly, the F1 score of count and tfidf are close to the baseline score. Even though we tried to simplify the tree into a smaller tree, but DT model is still overfitting from the observation of results. The reason for such poor performance is that the dataset is lack representative data which causes the decision tree to be too complex. Another disadvantage of the DT model is that it does not perform well with strong feature correlations. Glove100 with feature correlation capturing property got the highest score by recognizing relationships between features in the overfitting tree.

In the MLP result, all three features obtained high evaluation scores without complicated feature engineering. MLP is expected to reach better performance after further parameter tuning, but MLP has shown its effectiveness with limited parameter tuning. It is rare to see the count feature outperformed the other two features, the underlying reason for the outperformance is that neural networks can take "raw" data like the count feature as input and study them to construct models of nonlinear complex relationships. The reason that glove100 did not perform as well as the other two features is that MLP is sensitive to feature scaling, the vector mapping process in glove100 feature affected the studying of the MLP model.

In general, the combination of the LR model with the tfidf feature is 1% higher in F1 score comparing to the MLP model with the count feature. LR model has further improved its superiority and generalization ability in the natural language processing area. Thus, the LR model will be used to predict the test dataset.

5.2 Ethical Issues

An obvious bias that would significantly lower models' performance and generalization ability is language variation (Yang & Eisenstein, 2017). Language could change due to the laziness of humans; a popular example is that "laughing out loud" is constantly being written as "lol" online. People, especially the young generation tends to use the shorter version of words such as abbreviations. Not only abbreviation can simplify conversation and feeling expression, but it also becomes a trend to express personality. Models have difficulty distinguishing and classifying abbreviation, even though models could eventually classify the original word and its abbreviation into the same class, this process increases the workload and training time extensively.

6 Conclusion

In conclusion, this report assessed the effectiveness of logistic regression model, K-nearest neighbors model, decision tree model and multi-layer perceptron model on three different features "count", "tfidf" and "glove100". The logistic regression model with the tfidf feature slightly outperforms the multi-layer perceptron model with count feature in F1 score. However, there are plentiful parameter tuning works that can be done to the multi-layer perceptron model to improve its effectiveness. Future improvements include parameter tuning of multi-layer perceptron model and feature engineering with the existing features such as combining them as one feature.

References

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

Vadicamo, L., Carrara, F., Cimino, A., Cresci, S., DellOrletta, F., Falchi, F., & Tesconi, M. (2017).Crossmedia learning for image sentiment analysis in the wild. In2017 IEEE International Confer-ence on Computer Vision Workshops (ICCVW), pages 308-317.

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).

Kumar, A., & Sebastian, T. M. (2012).

Sentiment analysis on twitter. International Journal of Computer Science Issues (IJCSI), 9(4), 372.

Saif, H., He, Y., & Alani, H. (2012, November). Semantic sentiment analysis of twitter. In International semantic web conference (pp. 508-524). Springer, Berlin, Heidelberg.

El Kafrawy, P., Mausad, A., & Esmail, H. (2015). Experimental comparison of methods for multi-label classification in different application domains. International Journal of Computer Applications, 114(19), 1-9.

Moritz, P., Nishihara, R., & Jordan, M. (2016, May). A linearly-convergent stochastic L-BFGS algorithm. In Artificial Intelligence and Statistics (pp. 249-258). PMLR.

Zhang, M. L., & Zhou, Z. H. (2007). ML-KNN: A lazy learning approach to multi-label learning. Pattern recognition, 40(7), 2038-2048.

Karlik, B., & Olgac, A. V. (2011). Performance analysis of various activation functions in generalized MLP architectures of neural networks. International Journal of Artificial Intelligence and Expert Systems, 1(4), 111-122.

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

D´ıaz, M., Johnson, I., Lazar, A., Piper, A. M., and Gergle, D. (2018). Addressing age-related bias in sentimentanalysis. InProceedings of the 2018 chi conference on human factors in computing systems, pages 1–14.