COMP90049 Report: Does Unlabelled data improve twitter sentiment classification?

Anonymous

1 Introduction

Twitter sentiment classification has always been a popular topic to investigate in machine learning and so is the usage of unlabelled data. This report implements experiments with tweet data (Blodgett et al., 2016) to address report problem 1: Does Unlabelled data improve twitter sentiment classification? The results from the experiments were to be evaluated to find possible reasons for outcomes.

2 Literature review

Few papers were found to support the background knowledge for the report problem.

2.1 Benefit of unlabelled data

In the domain of sentiment classification, unlabelled data are relatively cheap and easy to obtain compared to labelled data, and the lack of labelled data is still one of the main challenging problems in sentiment classification (Yang, n.d.). Although unlabelled data does not contain class information, it can help to strengthen the knowledge about the joint distribution among features (Madhoushi et al., 2015).

2.2 Past approaches

To take advantage of the unlabelled data, people started to investigate different approaches to extract the useful features or data patterns from unlabelled data. Unsupervised learning was first invented, which required no label to do classification with techniques known as clustering. Nonetheless, the learning is often stuck due to the lack of knowledge about the dependencies of features and the sentiment classes (Aghababaei & Makrehchi, 2016).

2.3 Semi-supervised learning

Semi-supervised learning utilizes the given small size of labelled data to get the dependencies between features and labels and reaches out to the large set of unlabelled data (van Engelen & Hoos, 2019). Semi-supervised learning acts as the intermediate way between supervised and unsupervised learning, and it solves the task of lacking prior knowledge and makes use of the unlabelled data. It was the major method to explore the effect of unlabelled data in this report.

2.4 Data source

The raw Tweet data set used in the experiments was obtained from Blodgett et al. (2016).

3 Method

To test whether unlabelled learning can improve the performance within twitter sentiment classification, different models, evaluation methods and feature engineering methods were implemented.

3.1 Models

Several well-known models were used, including the k-nearest neighbour classifier, Naive-Bayes classifier, logistic regression classifier, and multi-layer perceptron classifier.

3.1.1 K-nearest neighbour classifier

K-nearest neighbour classifier classifies an instance by finding the nearest k data points and conducting a weighted formula. After the tuning tests, 5, and 20 were used as the k parameters, as 20 had the best performance of all from the tuning tests (Figure 1), while 5 is the most common k value. The 'Distance' weight method was chosen for the experiment since it weights the neighbours by the distance, which is an important factor when doing sentiment classification.

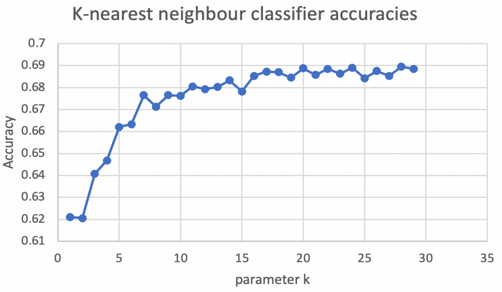

Figure 1- K-nearest neighbour classifier accuracies

3.1.2 Naïve-Bayes classifiers

Naive-Bayes classifier has the naive assumption that features of an instance are conditionally independent given the class. It is a probabilistic generative model which calculates join probability do classification.

3.1.3 Logistic regression classifier

A logistic regression classifier optimizes the conditional probability directly since it is a probabilistic discriminative model. Logistic regression uses the regression approach to calculate the probability of belonging to one class, and it utilizes a decision boundary to classify.

3.1.4 Multi-layer perceptron classifier

Multi-layer perceptron classifier constructs a neural network with different depths and widths. The depth of a neural network indicates the number of hidden layers it has, while the width indicates the number of neurons on each hidden layer. Multi-layer perceptron utilizes backpropagation to adjust the weight values among layers to make a better prediction in the next iteration.

After tuning tests, the best performance parameters without overfitting and underfitting were 'tanh' for activation function, 0.00005 for the learning rate, and one hidden layer with 8 neurons (Figure 2).

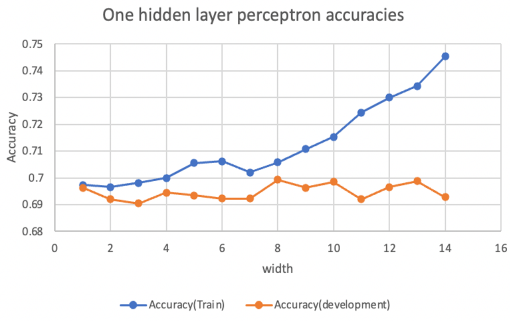

Figure 2- one hidden layer perceptron accuracies

3.2 Data sets

The raw Tweet data (Blodgett et al., 2016) was further divided into sets, and features engineered for the experiments.

3.2.1 Usage of data sets

The given data sets were separated into training data set and development data set for the training phase and evaluation phase respectively. It was like applying a holdout evaluation strategy which is simple and takes less time compared to other strategies.

3.2.2 TFIDF

The raw Tweet data (Schütze et al., 2008) was further feature engineered by the method term frequency-inverse document frequency pre-processing, which produced the data set that contains the 1000 highest TFIDF values based on the frequencies of the words.

3.2.3 Embedding vectors

Also, with the pre-trained language model, Sentence Transformer (Reimers and Gurevych, 2019), the raw data after feature engineer produced the 384-dimensional embedding vectors, which can capture words with the same meaning, hence shorter the distance between similar data.

3.3 Semi-supervised learning method

The semi-supervised learning model used in the experiment was the self-training model, which utilizes bootstrapping technique to enlarge the labelled data set with the most confidence among the unlabelled data set.

3.4 Evaluation measures

The experiments presented the results with accuracy and f-score for positive and negative classes. Other evaluation measures like contingency tables, precision and recalls are not presented for the simplicity of the report. Accuracy can illustrate the overall performance of the models while an f-score can show whether the data sets or the prediction is biased toward a certain class.

3.5 Baseline models

Zero-R (Zero rules) model was implemented as the baseline in this experiment, which is known as the majority class baseline. Zero-R classifies calculates the most common class in the training data and uses it in the future prediction.

The accuracy obtained from the Zero-R model with the tweet data set (Schütze et al., 2008) was 0.5.

4 Result

The results of the experiments were expressed in embedding data and TFIDF data versions.

Knn stood for the k-nearest neighbour models and the followed number was the k parameter.

Lr, cnb and mnb stood for logistic regression, complement Naïve-Bayes and multinominal Naïve-Bayes classifiers respectively.

Mlp was the multi-layer perceptron with a number indicating its number of neurons in the one hidden layer.

4.1 Embedding data

The one model testing results with embedding and TFIDF data sets.

	Accuracy	F-Score(P)	F-score(N)
knn(5)	0.662	0.661831	0.662169
knn(20)	0.68875	0.697742	0.679206
lr	0.69875	0.704294	0.692994
mlp(8)	0.69925	0.709911	0.687776

Table 1- supervised learning result (embedding)

	Accuracy	F-Score(P)	F-score(N)
knn(5)	0.6705	0.667675	0.673277
knn(20)	0.6775	0.690647	0.663185
lr	0.7005	0.707520	0.693135
mlp(8)	0.70125	0.708608	0.693511

Table 2- semi-supervised learning result (embedding)

4.2 TFIDF data

	Accuracy	F-Score(P)	F-score(N)
knn(5)	0.60875	0.597583	0.619314
knn(20)	0.614	0.596023	0.630445
lr	0.67675	0.675696	0.677797
mlp(8)	0.67575	0.672888	0.678563
cnb	0.6755	0.672553	0.678394
mnb	0.6755	0.672553	0.678394

Table 3- supervised learning result (TFIDF)

	Accuracy	F-Score(P)	F-score(N)
knn(5)	0.61325	0.624606	0.601186
knn(20)	0.6205	0.611367	0.629213
lr	0.67675	0.687304	0.665459
mlp(8)	0.67625	0.672069	0.680326
cnb	0.64725	0.645566	0.648918
mnb	0.647	0.672845	0.616721

Table 4- supervised learning result (TFIDF)

5 Discussion / Critical Analysis

In this section, the results from the experiments are discussed and analyzed along with the potential reasons leading to the results. Some further experiments were implemented to test the correctness of the discussion and analysis.

5.1 Learning performance

It is observed that unlabelled data has a boost effect on the performance of sentiment classification. Also, learnings with embedding data sets out-perform TFIDF data sets, nonetheless, the increment of accuracies between supervised and semi-supervised learning is larger within TFIDF data. Lastly, it is worth noting that the unlabelled data did not boost the two Naïve-Bayes and even decreased the accuracies.

5.1.1 Boost effect of unlabelled data

The boost effect of unlabelled data can be illustrated by the increment of accuracies between supervised and semi-supervised learning results in Tables 1 to 4. Furthermore, the increment of accuracies in learning with TFIDF data is more than in learning with embedding data (Table 3 and Table 4). The potential reason is that the relationships and joint distribution between features in TFIDF data require more data to be developed compared to embedding data, and the data was provided in the unlabelled data set.

5.1.2 Impacts of data sets features

Besides the boost effect, the learning with embedding data out-performed TFIDF data from Table 1 to Table 4. The reason is that TFIDF scores are counted based on the occurrence of words in Tweets (Yang, n.d.).

Words with the same meaning but different presentations are considered as different features in the TFIDF data sets, which separates the distance of data with the same sentiment even more. The phenomenon is significant in the comparison results of the k-nearest neighbour classifier (Table 1 and Table 3), which classifies data points highly depending on the distance. A person's speaking habits differ from one to another because of daily life, education levels and many other factors. Embedding data can shorter the distance between words with the same meaning, hence improving the performance.

5.1.3 Special Case: Naïve-Bayes

The accuracies in the semi-supervised learning for the two Naïve-Bayes classifiers have dropped compared to supervised learning (Table 4). It is caused by the 0 values existing in the TFIDF data sets, which affects the calculation process of the Naïve-Bayes algorithm. The situation got worsened due to the increment of the data size and the possible incorrect classifications.

5.2 High bias

High bias was observed in the results of supervised learning since the accuracies were no more than 0.7 (Table 1 and Table 3). The boost effects in semi-supervised learning suggested that unlabelled data can potentially solve the bias problem (Table 2 and Table 4). Besides unlabelled data, adding features is one of the ways to resolve the high bias problem, however, it is not possible in this experiment. Another way is to implement a more complex model, and it had been tried with multi-layer perceptrons with more depth and width, however, the problem of overfitting occurs. The other way is boosting, which combines multiple weak models to create a stronger model to reduce bias.

Supervised			
Models	Accuracy	F-Score(P)	F-score(N)

knn(5)
mlp(8)
lr
	0.7025	0.711165	0.693299

Semi-supervised			
Models	Accuracy	F-Score(P)	F-score(N)

knn(5)
mlp(8)
lr
	0.70275	0.709929	0.695206

Table 5- ensemble learning result (embedding)

Supervised			
Models	Accuracy	F-Score(P)	F-score(N)

knn(5)
mlp(8)
lr
	0.67625	0.674050	0.678421

Semi-supervised			
Models	Accuracy	F-Score(P)	F-score(N)

knn(5)
mlp(8)
lr
	0.676	0.673058	0.678890

Table 6- ensemble learning result (TFIDF)

With the further experiments, it is shown that not much improvement is observed between the results of the single model and ensemble regardless of learning rates and data sets being used (Table 5 and Table 6). Without other models to be tested with the ensemble method, it is hard to say whether the high bias problem comes with the models or the data sets.

5.3 Data distribution

Unbalanced data sets can potentially lead to worse performance in semi-supervised learning compared to supervised learning. Wrong classification information in the labelled data can later be exaggerated in the processing of classifying and enlarging data set with unlabelled data (Yang, n.d.). Nonetheless, it is not observed in the experiments, and it can be demonstrated that the distribution of classes in data sets is balanced with the following two test results.

5.3.1 Comparisons between Naïve-Bayes

Complementary Naïve-Bayes is adapted from the multinominal Naïve-Bayes to deal with data sets that have unbalanced distribution between classes. The two Naïve-Bayes classifiers in the supervised experiments had similar results (Table 3), indicating that the distribution of data sets was almost the same. This was further proved by counting the number of 'positive' and 'negative' in the data sets.

5.3.2 F-scores

In the experiment results of Table 1 to Table 4, there are no significant differences between the positive and negative F-scores, and it is an indication that the predictions of models were not inclined to any labels. However, the F-score for semi-supervised learning within the multinominal Naïve-Bayes classifier differed by 0.06 (Table 4), the situation should be because of the large number of zero values in TFIDF data sets. If adding more unlabelled data to the model, the inclination to the positive class is predicted to be worse.

5.4 Unexpected situation

Multi-layer perceptron was expected to perform the best among all the models since it was the only model that can tackle non-linear classification problems in the experiments.

However, the multi-layer perceptron did not out-perform other classifiers, which does not compile with the expectation. After further experiments, it was found that the potential reason was the complex requirements for the models vary for different demographic features.

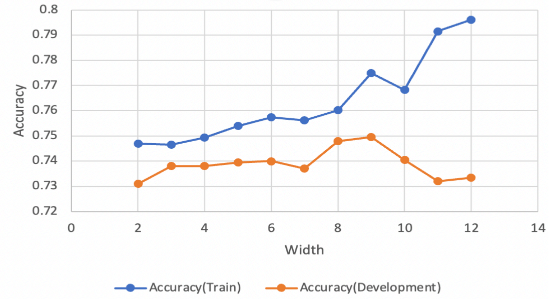

Figure 3- the accuracies with Standard American English

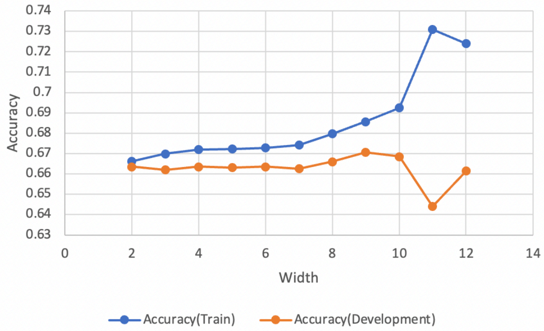

Figure 4- the accuracies with African American English

The Standard American English data required 8 neurons in the one hidden layer, on the other hand, the African American English data required 9 neurons, hence the advantage of the non-linear model might be eliminated.

6 Conclusions

From the experiment results, it is shown that unlabelled data does help to improve the classifier performance. semi-supervised learning successfully utilized unlabelled data to find joint distribution not shown in the labelled data set. Some worth noting numbers and results along with the unexpected situation and special case are also addressed with possible reasons. In future work, different unlabelled data should be tested to find the characteristics of a qualified unlabelled data set in the domain of sentiment classification.

References

Aghababaei, S., & Makrehchi, M. (2016, November 1). *Interpolative self-training approach for sentiment analysis*. IEEE Xplore; IEEE. https://doi.org/10.1109/BESC.2016.7804475

Blodgett, S. L., Green, L., and O'Connor, B. (2016). Demographic dialectal variation in social media: A case study of African-American English. *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.

Madhoushi, Z., Hamdan, A. R., & Zainudin, S. (2015, July 1). *Sentiment analysis techniques in recent works*. IEEE Xplore; IEEE. https://doi.org/10.1109/SAI.2015.7237157

van Engelen, J. E., & Hoos, H. H. (2019). A survey on semi-supervised learning. *Machine Learning*, *109*, 373–440. Springer Link. https://doi.org/10.1007/s10994-019-05855-6

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Schütze, H., Manning, C. D., and Raghavan, P. (2008). Introduction to information retrieval, volume 39.
Cambridge University Press Cambridge.

Yang, B. (n.d.). Semi-supervised Learning for Sentiment Classification. In *citeseerx.ist.psu.edu*. Department of Computer Science, Cornell University. Retrieved May 8, 2022, from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.403.339&rep=rep1&type=pdf