Report 2

Last update: April 30, 2019

1 Supervised Learning

We made seven attempts to improve the basic classifier. The first six attempts are based on changing the feature representation of texts and last one is to adjust the hyper-parameter of logistic regression.

1. Removing the words that are not Noun, Verb, Adjective, Adverb, Determiner or Modal verb. This attempt is based on the work of Turney (2002). In this work, the authors proposed five parts-of-speech (POS) combinations of adjectives and adverbs as sentiment features.
2. Using TF-IDF weighting. The basic classilfer uses TF weighting. Considering one more statistic for weighting the importance of features seems to bring higher accuracy.
3. Using boolean weighting. Boolean weighting means that the features are weighting by 0 or 1, indicating whether it appears or not. This attempt is based on the work of Pang et al. (2002). In this work, the authors proved that boolean weighting can outperform other weighting methods.
4. Using DF to reduce feature dimension. This attempt is based on the work of Wang and Zheng (2016). In this work, the authors proved that reducing feature dimension by DF can improve the accuracy. In our implementaiton, we set the min_df parameter in CountVectorizer() to remove the features with less DF than a threshold. The threshold is ranging from 0 to 0.5 with step value = 0.05.
5. Limiting the feature size. We use max_features parameter in CountVectorizer() to select the top N frequent words as features. N is ranging from 500 to 10,000 with step value = 500.
6. Using the combination of positive change on feature representation. We conducted the positive changes above in order to obtain a new feature representation.
7. Adjusting the regularization strength. After we obtain the optimal feature representation, we then search for the optimal regularization strength. We set C in LogisticRegression() ranging from 0.1 to 3 with step value = 0.1.

The benchmark labeled as 0 and results of first six attempts are listed in Table 1. The results of the fourth and the fifth attempts in the table are the highest accuracy when setting the optimal min_df and max_features respectively. The changes of accuracy when setting different min_df and max_features are presented in Figure 1 separately. From the results, we can find that the 1st, the 3rd, the 5th and the 6th changes have positive effect on performance. Using DF to reduce feature dimension is not suitable for the current dataset. The

optimal max_features is 6500. Figure 2 shows the change of accuracy with regularization strength. From the figure, we can find that the optimal regularization strength is 0.2 and the highest accuracy is 0.788.

Table 1: Result of the first six attempts

Attempts	0	1	2	3	4	5	6
Accuracy	0.777	0.779	0.766	0.788	0.777	0.782	0.788

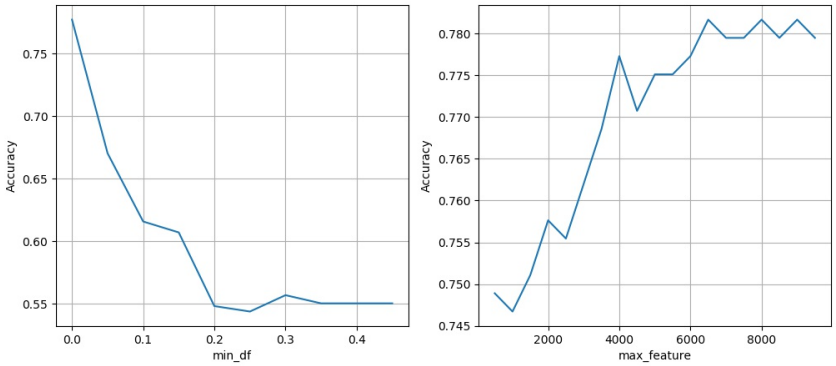

Figure 1: The Change of Accuracy with Varied min_df and max_features

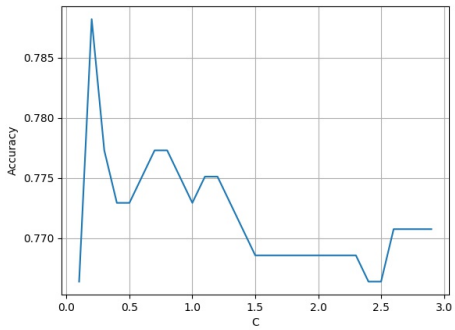

Figure 2: The Change of Accuracy with Varied Regularization Strength

2 Semi-supervised Learning

2.1 Expanding the Labeled Data

The diagram of the expanding process is as shown in Figure 3. The blue part shows how we training the classifier and predict the unlabled data, which is exactly the same as the supervised learning process. As show in the figure, we add top K confidently predicted samples into training data. In our experiment, we find that setting K as 0.1% of the unlabeled data enables us to obtain a better result than the supervised classifier. We

select the equal number of positive and negative samples in order to balance the training data. Furthermore, since the predicting probablity of logistic regression classifier indicate how certain it is to label the sample, we can regard this value as our prediction confidency. The idea that using the probabilities of sentiment classes to select samples is inspired by the work of Xiang and Zhou (2014). The stopping condition is that the minimun confidency of the new added subset is less than the threshold which we set as 0.98. In fact, we also set a stop point during the expanding process to locate the optimal performance among all iterations.

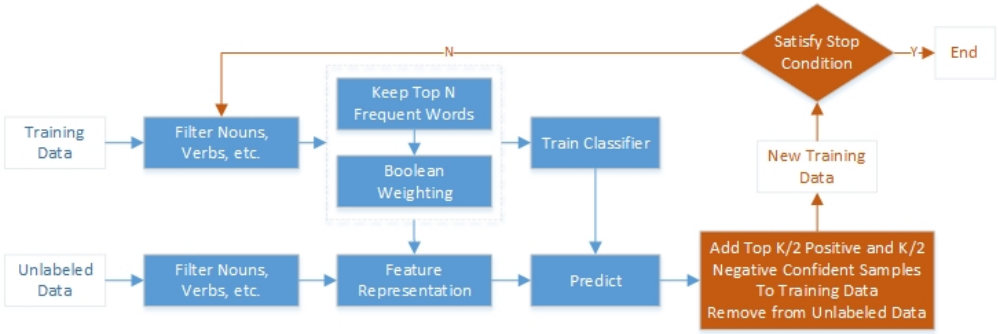

Figure 3: The Diagram of Expanding the Labeled Data

The change of performance on development data with the varied amount of training data is present in Figure 4. From the figure, we can find that the performance reaches the highest when the trainning data size is 5302, i.e. adding 720 predicted samples. After that, the accuracy is basically decreased. The highest accuracy is 0.79 which is slightly better than the benchmark result 0.788. From this result, we can infer that 1) adding a few samples that are very likely to be labeled correct has positive impact on performance because that more efficient training data usually brings about a better result, 2) adding too much predicted samples may introduce too much noise to the traning data, thus leading to the decrease of accuracy.

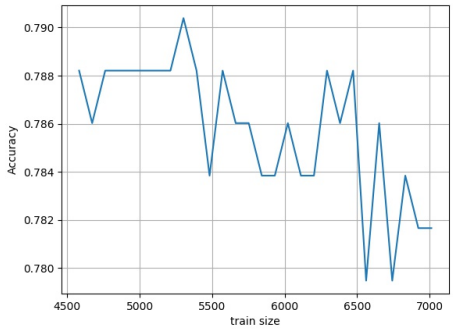

Figure 4: The Change of Accuracy with Varied Amount of Training Data

We list six examples of the highest and the lowest weighted features of the supervised classifier and the semi-supervised classifier in Table 2. The features are selected according to the rank of weights. We find that the three highest and the three lowest weighted features of the two classifiers are the same but their weights are different. From the result, we can find that the semi-supervised classifier assigns higher weights to the

three highest weighted features and lower weights to the three lowest weighted features than the supervised classifier. All the six features clearly expresses the positive or negative sentiment.

Table 2: Result of the first six attempts

Features	amazing	excellent	delicious	worst	horrible	terrible
SUP Weights	1.479	1.386	1.297	-1.791	-1.595	-1.177
SEMI Weights	1.514	1.407	1.334	-1.858	-1.668	-1.218

2.2 Designing Better Features

To solve the issue of sparsity, we can use deep learning method to learning the representation of contexts and words. For exmple, we can use the open source tool Word2Vec to learning the low-dimensional embeddings of words. Afterwards, the reviews/documents can be represented by the aggregation of the word embeddings. Then, we can obtain dense text representations for sentiment classification. However, this strategy relies heavily on the validity of word embeddings. When the unlabeled data are not enough for learning valid word embeddings, the performance will be even worse than the sparse represention.

3 Kaggle

- `Kaggle user name:` owenlovemika@gmail.com
- `Kaggle display name:` Owen Xu
- `email:` owenlovemika@gmail.com

References

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in "Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10" Association for Computational Linguistics 2002, pp. 79–86.

Turney, Peter D, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in "Proceedings of the 40th annual meeting on association for computational linguistics" Association for Computational Linguistics 2002, pp. 417–424.

Wang, Hongwei and Lijuan Zheng, "Sentiment classification of Chinese online reviews: a comparison of factors influencing performances," *Enterprise Information Systems*, 2016, *10* (2), 228–244.

Xiang, Bing and Liang Zhou, "Improving twitter sentiment analysis with topic-based mixture modeling and semi-supervised training," in "Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)," Vol. 2 2014, pp. 434–439.