Sentiment Classification via Product Attributes Discrimination

Mingyi Lu[1], Yuqian Cheng[2]

[1]*Department of Electrical Engineering, Stanford University*

[2]*Department of Computer Science, Stanford University*

Introduction

Sentiment analysis is a very popular task in natural language processing and social computing. Most studies usually consider reviews as individual and predict separately. However, the reviews should be connected through the product relations, since reviews of similar products tend to have similar polarity. As the attributes of product are very important factors to connect the reviews through similar products, we extract the attributes of each product and try to get common information from the review content or users for model training.

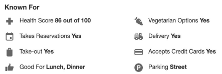
We propose a novel task to predict polarity of reviews jointly with restaurant attributes. We try on Naive Bayes and SVM models as they are standard models for classification tasks.

We also experiment on more complex models like Recurrent Neural Networks (RNN) and Long Short-Term Memory Networks (LSTM) to analyze the contribution of specific attributes.

Data and Evaluation

We conduct experiments on the dataset from the Yelp Dataset Challenge in 2013. We separate text reviews into positive and negative ones based on the stars, and select 4000 reviews as our dataset. Also we have a file with attributes of different restaurants, identified by business_id.

We define 80% data as training data, and the remaining 20% as testing data. We use precision, recall, and accuracy to evaluate the effectiveness of our models. Also for RNN and LSTM, we have mean-square error(MSE) measurement to better identify the contributions of different attributes.



The attributes we considered are attributes for a restaurant, as what are listed in the picture.

By comparing the performance of only reviews and reviews with several manually selected attributes, we want to know whether adding restaurant specific information will be helpful to identify the polarity of reviews. We append attributes information for each review by its corresponding restaurant, for example, “AcceptCreditCard_True”, then use either bag-of-words or other embeddings to process review input.

Methods

Naive Bayes

Our basic model is a Naive Bayes classifier, with bag-of-words feature extraction and Laplace Smoothing. We calculate the prior probability and conditional probabilities to get the maximum likelihood estimate.

The review is classified as 0 (negative) or 1 (positive) by:

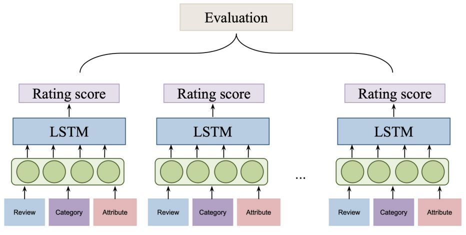
$$c_{\mathrm{NB}} = \underset{c \in C}{\arg\max}\, P(c) \prod_{i \in \mathrm{positions}} P(w_i|c)$$

Support Vector Machine (SVM)

SVM is a useful technique for classification tasks. As the input data is mapped to a high dimensional space, SVM tries to find the hyperplane that separates two classes and maximize the margin in the training procedure. The optimization problem is:

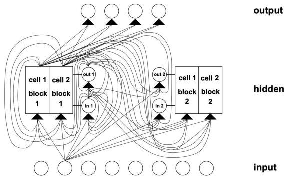
$$\min \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{s.t.} \quad y_i(\mathbf{w} \cdot \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i$$



r_1 r_2 r_n

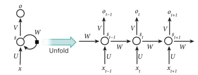
The architecture of our model



Long short-term memory structure

Recurrent Neural Network (RNN)

As our inputs are dependent of each other, we experiment on RNN as it could process sequential information. RNN takes sequence of input, and for every input, the output depends on previous computation. The input is each review and output is a predicted score.



Long Short-term Memory (LSTM)

LSTM model is also efficient in processing sequential data, and could solve the vanishing gradient problem encountered by RNN, as LSTM allows information to flow unchanged. Similar with RNN, for each review input, we use Word2Vec embedding with vector size 40, apply the LSTM layer with size 32, and apply sigmoid function to have a predicted score for each review between 0 (negative) and 1 (positive).

Results

Table 1. Test Result on Naive Bayes and SVM.

	Precision	Recall	Accuracy
Naive Bayes	0.9375	0.7440	0.8075
Naive Bayes w/ attributes	0.9350	0.7540	0.8150
SVM	0.8238	0.8300	0.8263
SVM w/ attributes	0.8213	0.8500	0.8388

For simple Naive Bayes and SVM, adding attributes leads to some improvement on the accuracy. As the precision doesn’t change much, we could infer that attributes could help us better analyze the negative reviews, as intuitively if a restaurant doesn’t have parking options, some negative reviews may be related to it.

Table 1. Test Result on basic RNN mode.

RNN	Price Range	Credit card	Wheelchair	Good for Groups	Parking Options	All attributes	No Attribute
Precision	0.8100	0.8175	0.8200	0.8225	0.7900	0.8075	0.7675
Recall	0.8329	0.8074	0.8283	0.8501	0.8144	0.8177	0.7832
Accuracy	0.8238	0.8113	0.8250	0.8388	0.8050	0.8138	0.7775
MSE	0.1331	0.1385	0.1296	0.1311	0.1359	0.1390	0.1664

Table 2. Test Result on LSTM mode.

LSTM	Price Range	Credit card	Wheelchair	Good for Groups	Parking Options	All attributes	No Attribute
Precision	0.8575	0.8300	0.8225	0.8075	0.8350	0.8800	0.8525
Recall	0.8265	0.8426	0.8613	0.8636	0.8586	0.8263	0.8158
Accuracy	0.8388	0.8375	0.8450	0.8400	0.8488	0.8475	0.8300
MSE	0.1219	0.1245	0.1188	0.1160	0.1115	0.1226	0.1301

From the results, we could see that adding individual and joined attributes are all effective improving the sentiment classification. Precision, recall, and accuracy all have improvements, and MSE for all experiments with additional information are smaller than the baseline. Some attributes have more significant contribution than others, like the price range of the restaurant, and whether it is wheelchair accessible.

Conclusion

The results on Naive Bayes, SVM, RNN and LSTM models support our hypothesis that, in sentiment classification task, restaurant attributes can be jointly considered with review text, as reviews are not only user specific, but highly related to the restaurant properties. The additional information leads to better performance of predicting the polarity of individual reviews.

REFERENCES

[1] Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 144-152). ACM.

[2] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273-297.

[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[4] Jurafsky, D., & Martin, J. H. (2008). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.

[5] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).

[6] Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*(Jul), 2121-2159.

[7] Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, *2*(3), 27.