Detecting negative reviews on Yelp
Machine Learning for Natural Language Processing 2020

Claudia Parziale
ENSAE Paris
claudia.parziale@ensae.fr

Karim Tit
ENSAE Paris
karim.tit@ensae.fr

Abstract

Sentiment analysis is part of the Natural Language Processing (NLP) techniques that consists in extracting emotions related to some raw texts. We consider the problem of classifying documents not by topic, but by overall sentiment, e.g., determining whether a review is positive or negative. So the goal of this project is to to explore how analytics can help businesses make decision-based on customer reviews. For our experiments we will use the Yelp Reviews Dataset, focusing on restaurants, to build a machine learning models which accurately predicts the sentiment of reviews. The code and main results can be found on the project's git [1].

1 Problem Framing

Each observation consists in one customer review for one restaurants. Each review is composed of a textual feedback of the customer's experience at and an overall rating from 1 to 5. We will add labels to this data corresponding to whether the review is considered positive or negative. More specifically a review will be positive it has 3 or more stars and negative otherwise. For the work described in this paper, we concentrated mostly on predicting negative sentiment. In order to simplify the problem we will remove from the dataset unnecessary columns.

We see that there is a big imbalanced, 80 of the reviews are positive and only 20 negative. There are different ways to handle this, however since this reflects the natural behavior of reviews, we decided to preserve the dataset bias.

As we can see in Figure n.1 most of the words utilized by the customers are indeed related to the restaurants: menu,food, place, etc. Some words are more related to the customer experience with the restaurant: amazing, great, super, unassuming, etc.

Whole idea here is that restaurant reviews are made of sequence of words and order of words encode lot of information that is useful to predict sentiment. Therefore we want to build a machine learning model with a high negative predictive value. To achieve this we will use three different embedding methods.

Figure 1: WordCloud from customer reviews

2 Experiments Protocol

First we propose to use the skip-gram word2vec embedding method, one of the most popular technique to learn word embeddings using a two-layer neural network. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space.

We will then train a Long short-term memory (LSTM). It is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data.

Our next step consists trying a more classic approach with a bag-of-words embedding, so to do

[1] https://github.com/karimtito/Machine_learning_NLP_Yelp

that we propose to use a CART algorithm, an attractive model if we care about interpretability. A decision tree is a supervised machine learning model used to predict a target by learning decision rules from features.

At the end, we will try train a bidirectional LSTM with it's own embedding layer. It is an extension of traditional LSTMs that can improve model performance on sequence classification problems. In problems where all time steps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence. The first on the input sequence as-is and the second on a reversed copy of the input sequence. This can provide additional context to the network and result in faster and even fuller learning on the problem.

To evaluate the performances of the different models, we decides to divide the reviews data set in a train and test set, and use respectively:

- **The false discovery rate:** The expected ratio of falsely claimed connections to all those claimed, is often a reasonable error-rate criterion in these applications.

$$FDR = \frac{TP}{FP + TP} \tag{1}$$

- **The negative predictive value:** In machine learning, the negative predictive value is defined as the proportion of predicted negatives which are real negatives. It reflects the probability that a predicted negative is a true negative.

$$NPV = \frac{TN}{TN + FN} \tag{2}$$

Let TP be true positives (samples correctly classified as positive,FN be false negatives (samples incorrectly classified as negative,FP be false positives (samples incorrectly classified as positive), and TN be true negatives (samples correctly classified as negative).

Once this is achieved our subsequent goal is to use the latter model to be able to give recommendations to businesses based on the words with the most predictive power. In particular this will be done using attention mechanisms, a recent trend in Deep Learning.

3 Results

In order to obtain more reliable results, we dividing our data in three sets, as is usual in data science : a training data set, a test set use to select hyper-parameters, and validation set to evaluate and compare the models. We see below that perfomances of the LSTM according to our metric is very poor. This might be due to not enough training data (we only used 15000 ligns because of time constraints). On the other hand a simple decision tree performs pretty well.

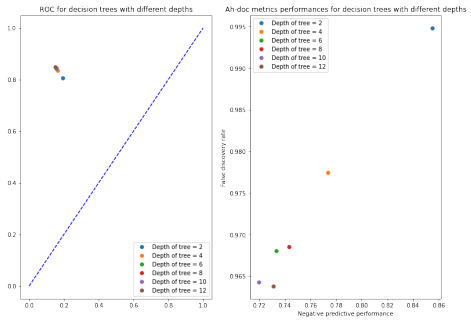

Figure 2: ROC and chosen metrics performance for Decision trees

Table 1: Results for the 3 models on validation data

Models	FDR	Negative predictive value
LSTM with Word2Vec	0.35	0.38
LSTM with embedding	0.20	0.21
Bag-of-word with CART	0.98	0.62

4 Discussion/Conclusion

Sentiment Analysis is an interesting way to think about the applicability of Natural Language Processing in making automated conclusions about text. It is completely possible to use only raw text as input for making predictions. The most important thing is to be able to extract the relevant features from this raw source of data. This kind of data can often come as a good complementary source in data science projects in order to extract more learning features and increase the predictive power of the models.

References

M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *CoRR*, abs/1402.3722.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.