Sentiment Classification of Reviews

Anaid Villamizar
Lancaster University
a.villamizaralbornoz@lancaster.ac.uk

1 Introduction

The increasing number of online reviews has become an interesting topic for sentiment analysis because it is feedback from the customers who have expressed their opinion or feelings about products, services, events or organizations. Therefore, reviews have important implications. Sentiment analysis of reviews is a powerful tool which helps companies to understand the customers, and make better decisions, suggest recommendations and more. Sentiment analysis is treated as classification problem in that the polarities reviews are classified (Haddi et al, 2013).

Our paper focuses on sentimental analysis of products, movies and restaurant reviews which classify the polarities reviews as positives or negatives (binary classification). Our research question is: How easily can the sentiment of reviews be predicted? To answer the question, we used 3 types of reviews, pre-processing and processing text methods, feature extraction method and classifiers.

The contributions of this paper are: We explored the use of 2 different pre-processing text methods, 3 different processing text methods, bag of word feature extraction method with neural networks and embedding of words with convolutional neural networks. Then we made the comparison between them and we studied the impact in the accuracy of each type of review.

This section will be followed by various sections: Section 2: The Related Work; Section 3: The methodology used; Section 4: Data used for the experiments; Section 5: The results of the experiments, Section 6: Findings about the results; Section 6: Conclusion and Future work.

2 Related work

In this section, we briefly review previous studies on sentiment classification.

There is a study where they explore the performance of Naive Bayes, Maximum Entropy and SVM in sentiment analysis of movies reviews. Those classifiers did not perform as well as other kinds of classification. The accuracy was between 75% and 83% (Pang et al, 2002).

There is another research where they compared three different machine learning algorithms to classify a large scale of online product reviews into positive and negative using high order n-grams as features and they found that discriminating the classifier combined with a high order n-grams as features performed better (Cui et al, 2006).

Sentiment classification of reviews in travel blogs have been done using 3 classifiers (Naïve Bayes, SVM and the character based N-gram model) and they found out that the SVM and N-gram approaches worked better than the Naïve Bayes (Ye et al, 2009).

There is a previous study where they used SVM to classify movies reviews and they explored the role of text pre-processing in sentiment analysis. They showed that using appropriate feature selection and representation of the performance significantly improvement (Haddi et al, 2013).

Additionally, there is another approach that consists of a simple convolutional neural network built on top of word2vec (pre-trained word vectors) and with less hyperparameter tuning. The experiments were carried out on different datasets including movie reviews where positive or negative reviews were detected (Kim, 2014).

Convolutional Neural Networks have been used for sentiment analysis. For instance, there is a research where they proposed a network called

Sentence Convolutional Neural (Dos Santos et al, 2014) with the word embeddings produced by unsupervised pre-training (Mikolov et al., 2013). They performed their experiments using movie reviews and Twitter messages. Their approach achieves a sentiment prediction accuracy of 86.4%.

Moreover, there is a paper wherein the authors carried out an empirical study on character-level convolutional networks for text classification which included sentiment classification of Amazon reviews. They compared difference methods and they found out that character-level ConvNet is an effective method (Zhang et al, 2015).

3 Data

We carried out the experiments using four review datasets such as: products, movies and restaurants. We took the datasets from the UCI Machine Learning Repository which provides 436 datasets to the machine learning community. The datasets contain 1000 reviews of products sold on amazon.com, 1000 Restaurant reviews from Yelp and 1000 movie reviews from IMDb. The label of these datasets is a binary variable; for positive sentiment it is 1, and for negative sentiment it is 0. Each type of the datasets contains 500 positive reviews and 500 negative reviews (Dua et al, 2017). The datasets are balanced.

We used these datasets because they are a subset of bigger real datasets. Moreover, to do sentiment analysis on reviews is very popular and these datasets have been used in other researches such as Kotzias et al (2015). Additionally, these datasets help to answer the research question using different 3 types of reviews.

Table 1 shows examples of products, movies and restaurant reviews.

Reviews	Polarity	Sentences
Products	Positive	The mic is great.
	Negative	Poor Talk Time Performance
Movies	Positive	His performance is simply genius
	Negative	I paid too much.
Restaurants	Positive	Wow... Loved this place.
	Negative	It's too bad the food is so damn generic.

Table 1. Examples of reviews

4 Methodology

4. 1 Pre-processing/cleaning the data

It is common that people make mistakes when they are writing reviews. Therefore, we corrected the spelling mistakes of the reviews. For this, we used Python 3 Spelling Corrector library (McCallum, 2016) to correct spelling mistakes and count the spelling mistakes. Moreover, we carried out experiments taking into consideration punctuations and without punctuations, and we compared the results of the classification.

4.2 Processing of the text method

We tokenized the reviews using nltk.word_tokenize function (Bird et al, 2009) and a customize tokenizer. The difference between them is that using customize tokenizer the emoticons are not separate in difference tokens. For example, each emoticon is considered as one token such as: ‘:)’, ‘:-)’, ‘:(’, ‘:-(’ are. Conversely, using nltk.word_tokenize function, the emoticons are lost, because there are separated as follows: ‘:’,’)’, ‘:’,’-‘,’)’, ‘:’,’(’, ‘:’,’-‘,’(’. Then we compared the results of the classification.

4.3 Feature extraction

We extracted features from the text. For this we used 2 extractor features:

- Bag of words (BoW) using Sklearn library (Jones et al, 2001) where the vocabulary size was 300 for each dataset and the vocabulary was different for each dataset.
- Embedding of words: we used GloVe: Global Vectors for Word Representation that is a pre-trained word vectors (Pennington et al, 2014). GloVe is similar to word2vec (Mikolov et al., 2013)., only GloVe is smaller (400,000 words). The dimensions of the embedding matrix were different for each data set.

To ensure fair comparison, the number of tokens were the same for BoW and embedding.

4.4 Classification

We built a neural network using Keras Library where the input was the bag of words. This neural network was based on Robinson et al, (2017). The neural network (NNs) consists of a dense layer wherein each neuron is fully connected to all neurons in the next layer. Then we used ReLU as an activation function. After that we added the last layer and the sigmoid activation function.

Additionally, we used a convolutional neural network (CNNs) based on Kera examples (2017). Each dataset was converted to numbers using a words list which was created with the unique tokens for each dataset. We built the convolutional neural network using Keras Library where the first layer used the embedding matrix. Then we added: a 1D convolutional layer of 1 dimension, a max pooling function, a hidden layer, ReLU activation function, the last hidden layer and the sigmoid activation function.

We trained each classifier with each dataset and then carried out the testing. We used 0.1 as a proportion of testing set.

4.5 Experiments

We carried out 18 experiments each of which we ran 10 times and then we calculated the mean of the accuracy. We divided the experiments into 3 groups. Table 2 shows each group of experiments.

Group	Pre-processing and processing text method
A	Punctuation + Customize tokenizer
B	Punctuation + nltk.word_tokenize function
C	Without punctuation + nltk.word_tokenize function

Table 2. Group of experiments

Each group of experiments was run using the neural network (input BoW) and convolutional neural network (input embedding).

5 Results

We carried out 18 experiments. We detected the number of incorrect spelling words in each dataset (Table 3).

Dataset	# Incorrect spelling
Products	573
Movies	837
Restaurants	702

Table 3. Number of incorrect spelling words by dataset

Then, we got the number of tokens by group of experiment. Table 4 shows the movie reviews have the highest number of tokens, those reviews are longer than other reviews.

Dataset	A	B	C
Products	12,160	11,955	10,201
Movies	14,910	16,646	14,294
Restaurants	11,228	12,678	10,847

Table 4. Number of tokens by datasets and token type.

Table 5 shows each of the emoticons that were found in the datasets during the group experiment A.

Dataset	Emoticons
Products	‘:).’ , ‘:-)’
Restaurants	‘:(‘ , ‘:)’
Movies	‘:)’, ‘:)’

Table 5. Emoticons

The accuracy can be observed in Table 6 for each dataset and group of experiments.

Group Exp.	Dataset	BoW (%)	Embedding (%)
A	Products	76.5	84.1
B	Products	**78.7**	**85.4**
C	Products	77.6	85.1
A	Restaurants	77.8	82.5
B	Restaurants	77.6	**83.5**
C	Restaurants	**78.5**	81.7
A	Movies	**73.4**	82.6
B	Movies	73.3	82.7
C	Movies	72.5	**83.3**

Table 6. Accuracy by experiments

In general, we have observed that using embedding+CNNs performed better than BoW+NNs (more than 5%).

In terms of preprocessing and processing text methods using BoW, we have different results for each dataset. For example, group experiment B using products dataset had better accuracy, group experiment C using restaurants dataset had the highest accuracy and group experiment A using movies datasets had the best accuracy. The pre-processing and processing text methods depended on the type of reviews. Therefore, with the movies dataset, it was important to create tokens with emoticons. However, compared to all other datasets, the movies datasets had the lowest accuracy.

In relation to pre-processing and processing text methods using embedding, the group of experiments that performed better were B and C. Hence, the extraction of the emoticons did not have an important effect.

5 Findings

- As we were expecting embedding + CNNs had better accuracy comparing to BoW+NNs as Zhang et al (2015). In general, the accuracy was better using any pre-processing and processing text methods. However, our results did not out performance recent studies. For instance, Dos Santos (2014) achieved sentiment prediction accuracy of 86.4% and we achieved 83.3% in movies revviews.
- The dataset with the lowest accuracy was movies. Other studies have mentioned that movie reviews are more difficult to classify (Pang et al, 2002), because people use different words to give their opinion about a movie compared to other kinds of reviews (Van et al, 2018). We had similar results as Pang et al (2002).
- The accuracy of sentimental analysis depends on different factors such as: pre-processing and processing text methods and type of reviews. So, some types of reviews can be easier to do sentiment analysis than others.

6 Conclusions and Future Work

In this paper, we focus on sentimental analysis of products, movies and restaurant reviews. Our experimental results show the pre-processing and processing text methods and type of reviews impacted the accuracy. Hence, the prediction of the sentiment of reviews could be easier in some types of reviews than another, because their characteristics are different. For example, movies reviews were more difficult to classify than restaurants and products.

In future work, we would like to try different pre-processing and processing text methods. For instance, we could exclude stopwords or remove some type of punctuation mark and improve the customize tokenizer.

7 References

Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O'Reilly Media Inc.

Cui, H., Mittal, V. and Datar, M. (2003) 'Comparative Experiments on Sentiment Classification for Online Product Reviews', pp. 1265–1270.

C. dos Santos and M. Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 69–78, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

Deshpande, A. (2017). Perform sentiment analysis with LSTMs, using TensorFlow. [https://www.oreilly.com/learning/perform-sentiment-analysis-with-lstms-using-tensorflow]

Dua, D and Karra T. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Haddi, E., Liu, X. and Shi, Y. (2013) 'The Role of Text Pre-processing in Sentiment Analysis', Procedia Computer Science. Elsevier B.V., 17, pp. 26–32. doi: 10.1016/j.procs.2013.05.005.

Jones E, Oliphant E, Peterson P, (2001). SciPy: Open Source Scientific Tools for Python. [http://www.scipy.org/]

Keras examples. [https://github.com/keras-team/keras/tree/master/examples]

Kim, Y. (2011) 'Convolutional Neural Networks for Sentence Classification'.

Kotzias, D. (2015) 'From Group to Individual Labels using Deep Features', pp. 597–606.

Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In Proceedings of Workshop at International Conference on Learning Representations.

McCallum, J. (2016). Python 3 Spelling Corrector. [https://pypi.org/project/autocorrect/]

Pang, B. et al. (2002) 'Thumbs up ? Sentiment Classification using Machine Learning Techniques', (July), pp. 79–86.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global Vectors for Word Representation.

Robinson, S., Gordon, J and Linhares, M. (2017). Intro to text classification with Keras: automatically tagging Stack Overflow posts. Google cloud big data and machine learning blog. [https://cloud.google.com/blog/big-data/2017/10/intro-to-text-classification-with-keras-automatically-tagging-stack-overflow-posts]

Van, V. D. (2017) ‘Combining Convolution and Recursive Neural Networks for Sentiment Analysis’, pp. 151–158.

Ye, Q., Zhang, Z. and Law, R. (2009) ‘Expert Systems with Applications Sentiment classification of online reviews to travel destinations by supervised machine learning approaches’, Expert Systems With Applications. Elsevier Ltd, 36(3), pp. 6527–6535. doi: 10.1016/j.eswa.2008.07.035.

Zhang, X., Zhao, J. and Lecun, Y. (2015) ‘Character-level Convolutional Networks for Text’, pp. 1–9.