Sentiment Analysis using Yelp: Does Noise affect the Accuracy of Algorithms?

Version 1.0

Omkar R Sunkersett, University of Michigan

Abstract

This research paper tries to solve the problem of sentiment analysis using Yelp reviews and help researchers decide the favorable machine learning techniques available to address the problem. It uses six supervised and two unsupervised learning methods to build sentiment predictive models, comparing their accuracies in the presence and the absence of noise within the reviews. It proposes two widely known supervised learning algorithms – logistic regression and linear support vector machines – in machine learning for solving the problem and provides further options to explore. The level of detail and comparison of different machine learning techniques provide researchers with a better understanding of the impact of noise on the accuracy of the algorithms, which are evaluated using the F1 score as the baseline.

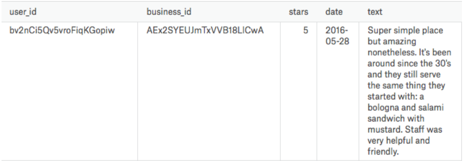
1 Introduction

It is difficult to decide a good machine learning technique to understand the sentiments of users on Yelp. Yelp has many customers (users) and businesses (service owners). Each customer has access to a variety of services offered by such businesses. Each customer can rate any service on a scale of 1 to 5. A rating of 5 indicates that the service is excellent, whereas a rating of 1 indicates that it is poor. Though such ratings are readily available, the users' sentiments are generally not known. This paper uses machine learning techniques to build predictive models to rate the users' sentiments as either positive or negative and compares the accuracy of these models in the presence and absence of noise for the given set of reviews.

This can help researchers determine the most favorable machine learning techniques for solving the problem of sentiment analysis on Yelp. The best techniques can be defined as those that produce a high overall accuracy both in the presence and the absence of noise for the given set of reviews. This paper also takes into account tuning each algorithm and could be used by researchers for performing further natural language analysis such as parts-of-speech tagging. It indicates that there are two machine learning techniques that stand out the most – logistic regression and linear support vector machines. This is because these two supervised algorithms retain their relatively high accuracy both in the presence and the absence of noise for the given data.

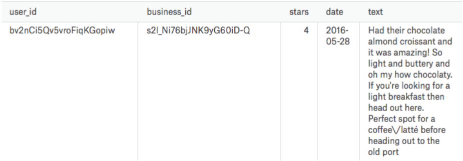
2 Problem Definition and Data

This research paper uses natural language processing techniques to perform sentiment analysis on a dataset obtained from Yelp. It analyses user reviews to determine whether they convey a positive or negative sentiment using supervised and unsupervised machine learning techniques. The supervised learning techniques that have been used are Random Forest, Bernoulli and Multinomial Naive Bayes, Linear Support Vector Machines, Logistic Regression and Stochastic Gradient Descent Russell and Norvig (2016). The unsupervised learning techniques that have been used are single-layered and multi-layered Perceptrons Hilton and Sejnowski (1999). The paper uses the F1 score Rijsbergen (1981) as an evaluation metric to evaluate four baselines: randomness retaining punctuation, hashtags, URLs & stopwords, default TF-IDF vectorizer parameters retaining punctuation, hashtags, URLs & stopwords, tuned TF-IDF vectorizer parameters retaining punctuation, hashtags, URLs & stopwords, and tuned TF-IDF vectorizer parameters removing punctuation, hashtags, URLs & stopwords. Particularly, it considers the following vectorizer parameters: minimum term frequency (min_df), maximum term threshold (max_df), N-gram range (ngram_range),



user_id	business_id	stars	date	text
bv2nCi5Qv5vroFiqKGopiw	AEx2SYEUJmTxVVB18LlCwA	5	2016-05-28	Super simple place but amazing nonetheless. It's been around since the 30's and they still serve the same thing they started with: a bologna and salami sandwich with mustard. Staff was very helpful and friendly.

Figure 1: Example of a positive review rated 5/5



user_id	business_id	stars	date	text
bv2nCi5Qv5vroFiqKGopiw	s2I_Ni76bjJNK9yG60iD-Q	4	2016-05-28	Had their chocolate almond croissant and it was amazing! So light and buttery and oh my how chocolaty. If you're looking for a light breakfast then head out here. Perfect spot for a coffee\/latté before heading out to the old port

Figure 2: Example of a positive review rated 4/5

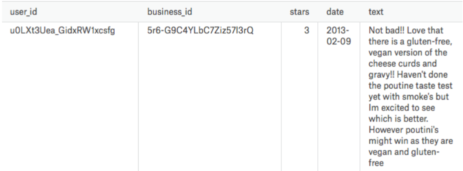
stopwords (stop_words) and lemmatization (tokenizer).

The Yelp dataset for this task has been obtained from Kaggle Yelp (2018). This dataset is 3 GB in size and contains 5,200,000 user reviews from 174,000 businesses across 11 metropolitan areas. It has many attributes amongst which those of interest are 'text' and 'stars'. These attributes represent the user-reviews and user-ratings of businesses, respectively. Additionally, this paper derives a new attribute called 'sentiment' to calculate the ground truth (1: positive sentiment, 0: negative sentiment). If a review has been rated ≥ 4, then the sentiment is considered as positive, else it is considered as negative (rated < 4).

For the scope of the project, the code randomly selects approximately only 10% of the entire dataset (600,000 rows) for training and testing purposes. This randomness helps us to ensure that the training and testing is performed on the data in an unbiased manner. It would be important to note that the number of positive reviews is twice that of the negative reviews. Therefore, it is necessary to randomly sample an equal number of positive and negative reviews from the original dataset. Further, the training to testing ratio for this activity is 70 to 30. This means that all the algorithms have been trained using 70% of the rows and then tested using the remaining 30%. Below are some examples of actual user reviews rated on a scale of 1 to 5 –

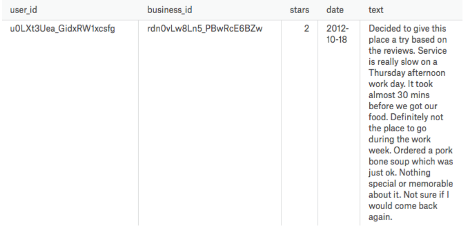
3 Related Work

The paper entitled "Aspect Based Recommendations: Recommending Items with the Most Valuable Aspects Based on User Reviews" Bauman et al. (2017) uses a sentiment utility logistic model to perform sentiment analysis of restaurants and recommend them, including their menu items. Similarly, the paper entitled "Sentiment Analysis and Subjectivity" Liu (2010) identifies facts and opinions from textual information and uses them as clues in the sentiment analysis task. Further, the paper entitled "Sentiment Analysis of Conditional Sentences" Narayanan et al. (2009) lays focus on identifying conditional sentences and their implications in the process of sentiment analysis. However, none of these papers take into account the noise present in the data. My paper considers this criteria and compares multiple machine learning algorithms to determine their accuracy in the presence and the absence of noise. It provides a wider range of supervised and unsupervised learning algorithms to address the problem of sentiment analysis on Yelp and proposes the most favorable machine learning techniques to address the problem.



user_id	business_id	stars	date	text
u0LXt3Uea_GidxRW1xcsfg	5r6-G9C4YLbC7Ziz57l3rQ	3	2013-02-09	Not bad!! Love that there is a gluten-free, vegan version of the cheese curds and gravy!! Haven't done the poutine taste test yet with smoke's but Im excited to see which is better. However poutini's might win as they are vegan and gluten-free

Figure 3: Example of a negative review rated 3/5

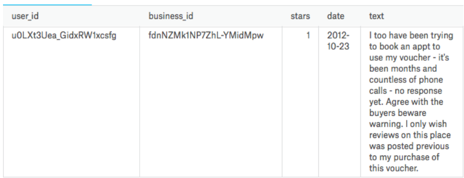


user_id	business_id	stars	date	text
u0LXt3Uea_GidxRW1xcsfg	rdn0vLw8Ln5_PBwRcE6BZw	2	2012-10-18	Decided to give this place a try based on the reviews. Service is really slow on a Thursday afternoon work day. It took almost 30 mins before we got our food. Definitely not the place to go during the work week. Ordered a pork bone soup which was just ok. Nothing special or memorable about it. Not sure if I would come back again.

Figure 4: Example of a negative review rated 2/5

4 Methodology

The research paper uses machine learning algorithms from the Python sklearn module on the given dataset. The first step is to load the entire



user_id	business_id	stars	date	text
u0LXt3Uea_GidxRW1xcsfg	fdnNZMk1NP7ZhL-YMidMpw	1	2012-10-23	I too have been trying to book an appt to use my voucher - it's been months and countless of phone calls - no response yet. Agree with the buyers beware warning. I only wish reviews on this place was posted previous to my purchase of this voucher.

Figure 5: Example of a negative review rated 1/5

dataset into memory and create a new attribute called 'sentiment', assigning a value of 1 to the reviews that have been rated ≥ 4 and a value of 0 to those that have been rated < 4. The Pandas sample() method is then used to randomly sample 300,000 positive reviews and 300,000 negative reviews from this dataset and randomly split the resulting dataset of 600,000 reviews across training and testing sets in the ratio of 70 to 30.

The next step is to decide each baseline. The first baseline is constructed by using the Numpy random.choice() method for predicting the sentiment in the reviews. This is the random baseline for the purpose of evaluation. Punctuation, hashtags, URLs and stopwords are retained during this baseline. The second baseline is built by using the Sklearn TfidfVectorizer() with default parameters, retaining punctuation, hashtags, URLs and stopwords. Whilst using this vectorizer, we can apply sublinear tf scaling, replacing tf with 1+log(tf), and also perform inverse-document-frequency reweighting. These tasks can be accomplished by setting both the 'sublinear_tf' and 'use_idf' vectorizer parameters to True. It is important to let the vectorizer know that each word should be vectorized by using the 'analyzer' parameter. The third baseline is then formed on top of the second baseline using the same TfidfVectorizer() but with tuned parameters, namely the minimum term frequency (min_df = 5), maximum term threshold (max_df = 95%), N-gram range (ngram_range = [2, 4]), English stopwords (stop_words = 'english') and lemmatization (tokenizer = LemmaTokenizer()). In this regard, LemmaTokenizer is a class defined using the NLTK WordNetLemmatizer. Finally, the fourth baseline is built similarly to the third baseline but removes punctuation, hashtags, URLs and stopwords from the reviews in order to check the net outcome on the performance.

The vectorizer is then transformed and fit onto the training data using the fit_transform() method to construct the training vectors before being transformed onto the testing data using the transform() method to build the testing vectors. A number of machine learning algorithms are trained, namely Random Forest, Bernoulli and Multinomial Naive Bayes, Linear Support Vector Machines, Logistic Regression, Stochastic Gradient Descent (supervised methods) and single-layered and multi-layered Perceptrons (unsupervised methods) Russell and Norvig (2016); Hilton and Sejnowski (1999). All of these algorithms use their default parameters. In particular, the multi-layered Perceptron uses 100 hidden layers by default. The training vectors and training labels are then given as input to each algorithm using the fit() method. The predicted labels are calculated using the predict() method on the testing vectors. Finally, the F1 scores are computed by comparing the predicted labels with the testing labels, using the binary average method (average = 'binary'). These scores are then tabulated for comparison purposes and the percentage of changes and differences are calculated Rijsbergen (1981).

5 Evaluation and Results

Below are the results for each baseline –

5.1 First Baseline: Randomness with Noise (table 1)

Table 1: Retaining punctuation, hashtags and URLs

Method	**F1 Score**
numpy.random.choice	0.499154

5.2 Second Baseline: Default Parameters with Noise (table 2)

Table 2: Retaining punctuation, hashtags and URLs

Classifier	**F1 Score**
Logistic Regression	0.891226
Stochastic Gradient Descent	0.886018
Linear SVC	0.885992
Multi-layered Perceptron	0.882945
Single-layered Perceptron	0.852125
Multinomial Naive Bayes	0.846515
Random Forest	0.756756
Bernoulli Naive Bayes	0.743506

5.3 Third Baseline: Tuned Parameters with Noise (tables 3 and 4)

5.4 Fourth Baseline: Tuned Parameters without Noise (tables 5, 6 and 7)

6 Discussion

From the results, we can infer that the first baseline (randomness) is poor with an accuracy of only 49.91%. Each machine learning technique

Table 3: Retaining punctuation, hashtags and URLs

Classifier	F1 Score
Multi-layered Perceptron	0.906628
Linear SVC	0.906292
Logistic Regression	0.901419
Multinomial Naive Bayes	0.892461
Single-layered Perceptron	0.891231
Stochastic Gradient Descent	0.885256
Bernoulli Naive Bayes	0.810029
Random Forest	0.796408

Table 4: Performance Change with Noise (wrt. default parameters)

Classifier	Percentage +/-
Bernoulli Naive Bayes	+8.9472
Random Forest	+5.2397
Multinomial Naive Bayes	+4.5946
Single-layered Perceptron	+4.5892
Linear SVC	+2.2912
Multi-layered Perceptron	+2.6823
Logistic Regression	+1.1437
Stochastic Gradient Descent	-0.0860

Table 5: Removing punctuation, hashtags and URLs

Classifier	F1 Score
Multinomial Naive Bayes	0.867349
Logistic Regression	0.864591
Linear SVC	0.860975
Multi-layered Perceptron	0.854645
Stochastic Gradient Descent	0.846590
Single-layered Perceptron	0.839866
Bernoulli Naive Bayes	0.820216
Random Forest	0.783318

Table 6: Performance Change without Noise (wrt. default parameters)

Classifier	Percentage +/-
Bernoulli Naive Bayes	+10.31733
Random Forest	+3.50998
Multinomial Naive Bayes	+2.46115
Single-layered Perceptron	-1.43864
Linear SVC	-2.82361
Logistic Regression	-2.98858
Multi-layered Perceptron	-3.20518
Stochastic Gradient Descent	-4.45002

beats this baseline. We can also observe that the tuned parameters (min_df = 5, max_df = 0.95, ngram_range = [2, 4] and tokenizer = LemmaTokenizer()) improve the overall accuracy of the classifiers. However, removing noise (punctuation, hashtags, URLs and stopwords) actually hurts the accuracy of most of the algorithms, especially the unsupervised algorithms such as the single-layered and multi-layered perceptrons for the given configurations. The Bernoulli Naive Bayes classifier is the only classifier that improves in accuracy by 1.26% on removing the noise from the given data. This helps us to surmise that the decision to remove noise from the given data should always take the machine learning technique to be used into account. The noise generally helps some of the supervised (logistic regression and linear support vector machines) and unsupervised learning techniques (single and multi-layered perceptrons) to improve in their overall accuracy.

Table 7: Performance Difference with and without Noise

Classifier	Percentage +/-
Bernoulli Naive Bayes	+1.2576
Random Forest	-1.6436
Multinomial Naive Bayes	-2.8138
Logistic Regression	-4.0856
Stochastic Gradient Descent	-4.3678
Linear SVC	-5.0003
Multi-layered Perceptron	-5.7337
Single-layered Perceptron	-5.7634

Moreover, fine-tuning the parameters of these algorithms helps us to improve their accuracy except for the Stochastic Gradient Descent. The accuracy of the algorithms increases due to the presence of noise. Once noise is reduced, the accuracy decreases for many of the algorithms except for the Bernoulli and Multinomial Naive Bayes and Random Forest algorithms. Particularly, the tuned Stochastic Gradient Descent algorithm is hurt the most in the absence of noise. On the contrary, the tuned single-layered and multi-layered perceptrons and supervised learning algorithms such as logistic regression and linear support vector machines improve in their overall accuracy. This happens because tuning helps reduce the error in these algorithms. The only exception to this observation is the Stochastic Gradient Descent, which decreases in accuracy in the absence of noise in spite of all the tuning effort. The best performing algorithms from the above results are logistic regression and linear support vector machines, which maintain a relatively high accuracy, retaining their position amongst the top-3 in the list both in the presence and the absence of noise for the given set of reviews.

7 Other Things I Tried

I tried to perform parts-of-speech (POS) tagging during lemmatization whilst training all of the algorithms, but the process had a huge time complexity since it was taking hours to train each algorithm with POS tagging during lemmatization. Hence, I decided to drop the idea and perform lemmatization without parts-of-speech tagging.

8 What You Would Have Done Differently or Next

I would try to build a parts-of-speech (POS) tagger having a lower time complexity and perform lemmatization with POS tagging whilst training the algorithms. It would be interesting to observe how POS tagging would affect the accuracy of each machine learning algorithm. I would anticipate a slight increase in the overall accuracy of each algorithm by performing parts-of-speech tagging.

9 Group Effort

This section is not applicable for me since it is an individual project.

Acknowledgments

I would like to thank Professor David Jurgens and the instructional team for guiding us through such a fantastic course. The lectures, reading material, questions and assignments helped me to understand the core concepts of natural language processing, especially sentiment analysis. Further, this project has let me reinforce these learnings by enabling me to experiment with them by using real-world data.

References

Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. 2017. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. ACM, New York, NY, USA, pages 717–725.

Geoffrey Hilton and Terence J. Sejnowski. 1999. *Unsupervised learning: foundations of neural computation*. Massachusetts Institute of Technology.

Bing Liu. 2010. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing, Second Edition. Taylor and Francis Group, Boca*.

Ramanathan Narayanan, Bing Liu, and Alok Choudhary. 2009. Sentiment analysis of conditional sentences. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*. Association for Computational Linguistics, pages 180–189.

C. J. van. Rijsbergen. 1981. *Information retrieval*. Butterworth.

Stuart J. Russell and Peter Norvig. 2016. *Artificial intelligence: a modern approach*. Pearson.

Inc Yelp. 2018. https://www.kaggle.com/yelp-dataset/yelp-dataset.

A Supplemental Material

The Python code for this project is present in the file called 'si630_project_code.pdf'. It is compatible with Python 3.6+.