SENTIMENT ANALYSIS USING NATURAL LANGUAGE PROCESSING

Avish Kadakia 1132452 | **Dhyan Shah** 1131707 | **Prithvisinh Jhala** 1137435 | **Vikas Patel** 1141148 | **Sushmita Darimadugu** 1149657

Abstract

In an E-Commerce (EC) context, reputation and trust are extremely important and play a critical role in enabling numerous parties to form mutually beneficial interactions. In terms of reputation, a number of variables have a detrimental impact on customers' and sellers' perceptions. For example, users may generate phantom feedback reviews to maintain their reputation due to a lack of trustworthiness in delivering feedback reviews. As a result, we will believe that the reviews and ratings are unjust. We have used Sentiment Analysis in our project, to detect unfair negative reviews, unfair neutral reviews, and unfair positive reviews. A collection of consumer reviews is utilised to test sentiment categorization systems. We compare three supervised machine learning methods for sentiment classification using a dataset of amazon product reviews: KNN, SVM(SGD Classifier), and Multinomial Naive Bayes. We have then used accuracy, precision, F1 score, and recall as performance measures in order to assess sentiment categorization performance. When compared to the other three classifiers, our trials reveal that the SVM algorithm is the best and most accurate in conjunction with LSA as the word embedding technique.

1 INTRODUCTION

Increasing use of the internet makes a lot of structured and unstructured data available online. Unstructured, text-based data makes up as much as 80 percent of the data now available on the Internet. This data is continually being created from a wide range of sources, including email, text messages, blogs, social media posts, and product reviews. There is a huge amount of data that needs to be structured and processed to get insights, those insights help in developing businesses. A study on amazon in 2017 revealed over 88 percent of online shoppers trust reviews as much as personal recommendations.

For example, Amazon is one of the e-commerce giants that people are using every day for online purchases where they can read thousands of reviews dropped by other customers about their desired products. Analyzing this enormous number of opinions is also hard and time consuming for product manufacturers. Amazon has attempted to address this issue by emphasizing the "Most Helpful Customer Reviews." This solution only reflects a small portion of the population and can be improved with more advanced data analysis. With this solution Amazon can give better suggestions to people related to their purchase history, which eventually helps in growing business[1].

In this paper we have worked on sentiment analysis for Amazon product reviews. We have performed literature review regarding the same as well. Here we have chosen Amazon wireless products dataset, followed by our approach for the project. We have used 3 different word embedding techniques and 3 different classifiers. Detailed methodology for better understanding of our approach is provided in the future sections of this paper.

2 LITERATURE REVIEW

Sentiment analysis has a gained a lot of attention in recent years due to advances in the field of NLP. For our literature review we have analysed various previous approaches for sentiment analysis on datasets using Multinomial Naive Bayes MNB, Support Vector Machine SVM, Long Short-Term Memory LSTM etc.

2.1 Sentiment analysis of Yelp's ratings based on text reviews

Y.Xu, X.Wu, and Q.Wan. from Stanford University worked on existing supervised learning algorithms such as perceptron algorithm, naive bayes and support vector machine to predict a review's rating. They have used data scraping from amazon url to get the data and preprocessed it. They have distributed their dataset as 70% as training data and 30% testing data. They have used different classifiers to determine the precision and recall values. Naive Bayesian and decision list classifiers were used to classify a given review as positive or negative. They found that binarized Naive Bayes combined with feature selection with stop words removed and stemming is the best suited for their problem set[2].

2.2 Amazon Reviews, business analytics with sentiment analysis.

Elli, Maria Soledad, and Yi-Fan Wang extracted sentiments from the reviews and analyzed the results to build up a business model. They mainly used Multinomial Naïve Bayesian (MNB) and support vector

machines (SVM) as classifiers. Their model is a supervised learning method that uses a mix of 2 kinds of feature extractor. They have achieved an accuracy of over 90% with the F1 measure, precision and recall. In most of the cases 10 fold provided the best accuracy while Support Vector Machine (SVM) provided best classification results[3].

2.3 Sentiment Analysis in Hotel Reviews Based on Supervised Learning.

Han-xiaoshi, Xiao-jun in "Sentiment Analysis in Hotel Reviews Based on Supervised Learning." proposes a supervised machine learning approach using unigram features and TF-IDF. They extracted sentiments from the reviews in which each instance in the training set contains one target value or class label and several attributes/features. The goal of a support vector machine (SVM) is to produce a model which predicts the target value of data instances in the testing set which are given only the attributes. SVM has been shown to be highly effective at traditional text categorization, generally outperforming Naive Bayes. They are large-margin, rather than probabilistic, classifiers, in contrast to Naive Bayes[4].

2.4 Learning user and product distributed representations using a sequence model for sentiment analysis

T.Chen, R.Xu, Y.He, Y.Xia, and X.Wang have shown working on sequence modeling based neural network for document-level sentiment analysis. They have used a one-layer convolutional neural network (CNN) which is used to learn review embeddings of IMDB and Yelp datasets where they have used reviews of varying length and produced 300-dimensional vectors. Shorter reviews were padded with zero vectors to accommodate various duration's of reviews. Sentiment filters of 3 and 5 widths were applied to the word embeddings to perform one-dimensional convolution and generate numerous feature maps.

Useful characteristics were collected by using max-overtime pooling in the pooling layer. Multiple filters' outputs are then concatenated to form a 300-dimensional vector. To train the network over K-classes, the softmax function is utilised as an activation function. They then employed a Recurrent Neural Network (RNN) to learn the temporal information as well as to capture both product and user information, and they have claimed cutting-edge results on the IMDB and Yelp datasets[5].

3 DATASET

We have focused on automatically predicting ratings for Amazon Wireless product reviews using user review data. We have chosen the Amazon Wireless review dataset for our problem statement as it has over 8 million labeled wireless product reviews by customers. This dataset consists of reviews from the duration of May 1996 to October 2018. The first step in our research was to import the data for preprocessing. The data was provided by Amazon[6].

The data set consists of a total of 8,991,589 rows/reviews of different wireless products. Each review has a star rating from 1 to 5 where (1) is the worst and, (5) is the best. For our experiment we have performed 2 types of sentiment analysis, the first one includes 5 sentiments from 1 to 5 ratings and in the second analysis we have grouped ratings as 1 - 2 (negative), 3 (neutral) and 4 -5 (Positive). There are a total of 17 columns in the original dataset but we discarded most of the columns and have used the "reviewbody" column to perform sentiment analysis and the "starrating" column as our labels for training the models.

No imbalance of classes was observed in our dataset analysis. After preprocessing, our data has been divided into testing (33%) and training (66%) to compare performance difference between the various approaches.

4 OUR APPROACH

We have extracted features from the dataset after preprocessing in the form of a word vector using the feature extraction algorithms CountVector, LSA encoding, Word2Vec embedding. Next, we have classified these embedded vectors using KNN, Support Vector Machine(SVM), and Multinomial Naive Bayes. We have tested the algorithm with cross-validation and calculated Precision, Recall, F1-score. We have also calculated these results for these 2 conditions:

- 5 classes for ratings 1,2,3,4,5
- 3 classes for ratings (1-2), (3), (4-5)

The following figure provides a description of our system. We have tried different combinations of word embedding techniques and classifiers to achieve the best results with the highest accuracy.

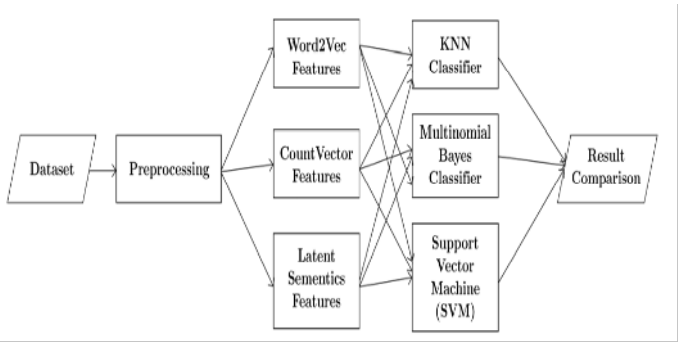

Fig. 1. System Overview Diagram

5 METHODOLOGY

For our predictions we will be following the standard practice for working with text data consisting of the following steps (Fig. 1):

- Importing Dataset
- Data pre-processing
- Word Embedding / Feature Extraction
- Making predictions using classifiers

As importing dataset and preprocessing are self-explanatory let us move to the more complicated parts of the implementations.

6 FEATURE EXTRACTION

When it comes to developing any sort of prediction algorithm it is always best to work with numerical data as we can perform a plethora of mathematical operations and extract valuable information from them. The various techniques which are used to convert textual data to numerical data are called Feature Extraction or Word Embedding. For our use case we have chosen the following 3 methods:

6.1 Word2Vec

Word2Vec is a very well known technique in the NLP field for feature extraction, Introduced by Tomas Mikolov at Google in 2013. It uses a simple one-layer neural network to perform feature extraction and gives us a final representation of words in the form of a vector. It has 2 Models, Continuous Bag-ofWord(CBOW) Model, and Skip-Gram Model. The CBOW model predicts the current word based on context while the SkipGram model predicts surrounding words given the current word.

For our approach we have extracted one hot encoded representation for all the words. We have then passed these vectors as input to a one-layer feed-forward network. The neural network then uses softmax as it's activation function to train and update it's weights using a gradient descent algorithm. We then get our final word vector at the end of the training process. Using which we have been able to train our classifiers and generate predictions [7][8].

6.2 Latent Semantic Analysis

Using this technique we have tried to extract the Latent semantics i.e. hidden features that represent something essential in our dataset. It's an unsupervised technique. Once we have our counter vector we have also used various dimension reduction techniques such as SVD. LSA is divided into 2 main parts, Generating Document Term matrix (Fig. 2) and Performing Singular Value Decomposition on that matrix to get 3 matrices. After performing SVD we have spliced the matrix and have gotten the first few columns and combined all matrices again so that we can easily calculate cosine similarity between words. This similarity has then been used by the classifiers to make more accurate predictions as it will also encode the semantic similarity between words [9][10].

	brown	dog	fox	lazy	quick	red	slow	the	yellow
"the quick brown fox"	1	0	1	0	1	0	0	1	0
"the slow brown dog"	1	1	0	0	0	0	1	1	0
"the quick red fox"	0	1	0	0	1	1	0	1	0
"the lazy yellow fox"	0	0	1	1	0	0	0	1	1

Fig. 2. Document Term Matrix

6.3 CountVector

CountVectorizer is a fairly simple algorithm that works on Term Frequency, i.e. counting the occurrences of tokens and building a sparse matrix of document X tokens. Once we have the count vectors we have then been able to apply various classifiers on the matrix to generate predictions. CountVectors is a very basic technique to convert words to vectors as it does not take into consideration any relation the words might have with each other. But even this basic method has given us a decent accuracy boost rather than directly working with textual data.

7 CLASSIFIERS

Our final goal is to predict user sentiment between 1-5(1 being worst and 5 being best). As this is a classification problem we have used various multiclass classifiers in order to get the predictions. The implementation for the various classifiers we will be comparing are as follows:

7.1 KNN (K Nearest Neighbours)

KNN is one of the most simple and traditional non-parametric techniques to classify samples. This algorithm is also called a lazy learning algorithm since it does not need any training data points for model generation. All training data is used in the testing phase. This makes training faster and the testing phase slower and costlier. Given an input vector, KNN calculates the approximate distances between the vectors and then assigns the points which are not labeled to the class of its K-nearest neighbors. We have used unsupervised learning and have created clusters based on the number of labels. Then we have used MiniBatchKMeans() to generate small random batches of data of a fixed size for each iteration, a new random sample from the dataset is also obtained to update the clusters. Then partialfit() is used to train the model incrementally with each batch iteration. Finally testing

is performed and accuracy is calculated. Classificationreport() is used to display testing results including precision, recall, F1, and support scores for the model.

7.2 Multinomial Naive Bayes

$$\Pr(c|t_i) = \frac{\Pr(c)\Pr(t_i|c)}{\Pr(t_i)}, \qquad c \in C$$

Fig. 3. Bayes Theorem

In Multinomial Naive Bayes, we have calculated the probabilities of each word for a given document, using the Bayes formula (Fig. 4). We have then calculated the highest probability and assigned that class to a new document. We have then calculated priors Pr(c) by dividing the number of documents belonging to class c by the total number of documents and also calculated the rest of the terms in formula. The approach is quite straightforward in theory but one of the issues is that it does not take into account the similarity between words. This is one of the reasons why it is referred to as a Naive approach. The predictions can still have high accuracy irrespective of this drawback. As can be seen in the research paper "Multinomial naive bayes for text categorization revisited" [11] and our results.

7.3 Support Vector Machine (SVM)

SVM is a supervised learning model that has a solid theoretical foundation and performs classification more accurately than most other algorithms in many applications. Many researchers claim that this is the best text classifier and can be used for sentiment analysis to get great performance. SVM linearly separates data and finds a hyperplane that fits best between 2 classes but when data is not linearly separable then we can convert data to a higher dimension and hope to find a hyperplane in a higher dimension, we can do this using kernel functions [12]. We can also see that SVM in conjunction with LSA gave us the highest accuracy in both cases in the result section.

8 RESULTS

The results of CounterVector, Word2Vec and Latent Semantic Analysis word embedding techniques with all 3 classifiers for both 5 and 3 sentiments are shown below:

- 3 Classes / Ratings (1-2), (3), (4-5) :

	CounterVector	Word2Vec	LSA
KNN	0.3500	0.7024	0.4310
MNB	0.6926	0.6523	0.7024
SVM	0.6721	0.6915	0.7873

- 5 Classes / Ratings 1,2,3,4,5 :

	CounterVector	Word2Vec	LSA
KNN	0.1442	0.5355	0.6064
MNB	0.5246	0.5109	0.5355
SVM	0.5072	0.4867	0.6064

As per our experimentation results, it can be clearly observed that more than the transformation techniques and classifiers used for sentiment analysis the number of classes has a much higher impact on the accuracy especially for the dataset we have used. It is also reasonable to conclude for the dataset using a combination of SVM with LSA as a feature extractor provides us the best results.

It is interesting to observe that KNN had the worst performance which was significantly worse than the other classification methods. As per our understanding this may be because KNN being an unsupervised learning algorithm did not have access to the ratings while training which could have made it very difficult for the model to create the clusters accurately.

9 CONCLUSION AND FUTURE WORK

Experimental results indicate that training on written product reviews is a promising alternative to exclusively using (spoken)in-domain or user rating data for building a system that analyses product reviews. We have implemented the major steps of data importing and cleaning. We have also implemented all the classifiers and word embedding techniques (Word2Vec, LSA, CounterVector) and generated the other results for 3 and 5 sentiments.

Comparing all of these performances we received better accuracy for 3 sentiments rather than 5 sentiments. Moreover Multinomial Naive Bayes and Support Vector Machine provided better results rather than KNN. While comparing between Multinomial Naive Bayes and Support Vector Machine, Support Vector Machine (SVM) gave better results when implemented with Latent Semantic Analysis word embedding technique.

We have also seen a significant rise in accuracy as the number of classes decreased. There may be multiple reasons for such a performance boost, but as per our understanding, this may be because even as humans if we are given a review with a rating of 5 and asked to re-rate it, we may just as easily classify it as a 4 rating rather than a 5 rating.

This is because providing a rating for a review can be quite arbitrary and is heavily influenced by an individual's opinion and cannot be predicted, as 2 people who may be given the same review may rate them differently based on their opinions. This makes it quite hard to find a determinable pattern to rate reviews for a classifier.

As a result, even the algorithms that we currently use we cannot guarantee high accuracy. The current methods are best at approximating at a high level if a review is positive, negative or neutral rather than predicting the exact rating for the reviews. As it is clearly evident from our results even though the same embedding techniques and classifiers were used, the accuracy we achieved for 3 classes and 5 classes vary drastically. It is also reasonable to conclude that the dataset using a combination of SVM with LSA as a feature extractor provides us the best results. We also noticed that KNN with CountVector gave us the worst result.

While the algorithmic approach using Multinomial Naive Bayes and svm is surprisingly effective, It suffers from 3 fundamental flaws:

- The algorithm produces a score rather than a probability.
- The algorithm 'learns' from examples of what is in a class, but not what isn't.
- Classes with disproportionately large training sets can create distorted classification scores, forcing the algorithm to adjust scores relative to class size.

As it is quite evident the biggest issue is the decrease in accuracy as the number of classes increases. So for the future, we may want to train on an even larger dataset using machine learning models. This is because given a large enough dataset, machine learning models like ANNs, CNNs, LSTMs etc. may be able to learn various patterns which may help them better predict the ratings for a review.

10 REFERENCES

[1] Chantal Fry and Sukanya Manna. 2016. Can we group similar amazon reviews: A case study with different clustering algorithms.

[2] Wang, Q., Wu, X., Xu, Y. (2016). Sentiment Analysis of Yelp's Ratings Based on Text reviews. http://cs229.stanford.edu/proj2014/Yun Xu, Xinhui Wu, Qinxia Wang, Sentiment Analysis of Yelp's Ratings Based on Text Reviews.pdf. Accessed 20 October 2016.

[3] Elli, Maria Soledad, and Yi-Fan Wang. "Amazon Reviews, business analytics with sentiment analysis." 2016. https://docplayer.net/151407565-Amazon-reviews-business-analytics-with-sentiment-analysis

[4] Han-xiao shi,Xiao-ju. Sentiment Analysis in Hotel Reviews Based on Supervised Learning. Internationals Conference on Machine Learning and Cybernetics.

[5] Chen, T., Xu, R., He, Y., Xia, Y., and Wang. X. Learning user and product distributedrepresentations using a sequence model forsentiment analysis. https://core.ac.uk/download/pdf/78899161.pdf.

[6] https://s3.amazonaws.com/amazon-reviewspds/tsv/index.txt.

[7] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[8] Rong, Xin. "word2vec parameter learning explained." arXiv preprint arXiv:1411.2738 (2014).

[9] Deerwester, Scott, et al. "Indexing by latent semantic analysis." Journal of the American society for information science 41.6 (1990): 391-407.

[10] Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. "An introduction to latent semantic analysis." Discourse processes 25.2-3 (1998): 259-284.

[11] Kibriya, Ashraf M., et al. "Multinomial naive bayes for text categorization revisited." Australasian Joint Conference on Artificial Intelligence. Springer, Berlin, Heidelberg, 2004.

[12] Moraes, Rodrigo, JoaO Francisco Valiati, and Wilson P. Gavi ˜ aO Neto. ˜ "Document-level sentiment classification: An empirical comparison between SVM and ANN." Expert Systems with Applications 40.2 (2013): 621-633.

11 APPENDIX

A. List of Figures:

B. Contributions of team members.

All the team members of group 14 have decided to work on this project mutually. Following is a breakdown of individual contribution of the group mates and their future planned work.

Avish Kadakia
Role: Team Lead

- Condensing and coordinating information between the various team mates.
- Implementing the various classifiers and embedding techniques along with Prithvisinh Jhala and Sushmita Darimadugu.
- Proof reading and final edits for project proposal and project progress report.
- Creating project approach and finalizing classifiers for implementation.
- Making final edits for the project presentation and report.
- Aggregating the results and generating insights for report and presentation.

Dhyankumar Shah
Role: Database Analysis and Assistant Programmer

- Selecting the database and researching the topic.
- Importing and cleaning dataset along with Vikas Patel.
- Creating the Project proposal literature review and problem statement along with Vikas Patel.
- Creating and formatting slides for the project presentation.
- Preforming literature review for the final project proposal.

Prithvisinh Jhala
Role: Lead Programmer I.

- Formatting and Editing the project proposal and finalizing various word embedding techniques.
- Creating methods to pre-process the imported datasets to improve classifier accuracy.
- Implementing the various classifiers and embedding techniques along with Avish Kadakia and Sushmita Darimadugu.
- Implementing remaining word embedding techniques with Sushmita Darimadugu. Generating visualizations for the final results for the project presentation and final report

Sushmita Darimadugu
Role: Lead Programmer II.

- Formatting and Editing the project progress update report.
- Creating methods to run the various classifiers in conjunction with the embedding techniques.
- Implementing the various classifiers and embedding techniques along with Avish Kadakia and Prithvisinh Jhala.
- Implementing remaining word embedding techniques with Prithvisinh Jhala.
- Condensing all the results and generating insights for the project presentation and final report.

Vikas Patel
Role: Research Expert and Assistant Programmer

- Importing and cleaning dataset along with Dhyankumar Shah.
- Creating the project proposal literature review and problem statement along with Dhyankumar Shah.
- Creating and formatting project proposal sections and designing various approach diagrams.
- Performing in-depth analysis for final project report. Condensing all data, insights and result for final report and presentation.