Sentiment Analysis of Amazon Reviews

1st Roman Bellisari
Stevens Institute of Technology ECE
Computer Engineering
Hoboken, NJ
rbellisa@stevens.edu

2nd Matthew Lepis
Stevens Institute of Technology ECE
Computer Engineering
Brick, NJ
mlepis@stevens.edu

3rd Michael Delcid
Stevens Institute of Technology ECE
Electrical Engineering
West New York, NJ
mdelcid@stevens.edu

***Abstract*—The rapid growth of e-commerce has presented companies with an enormous volume of online reviews for products. These reviews often come in one of two formats: numeric ratings or textual comments on the product, or a combination of both. One of the biggest challenges both companies and consumers face is determining, from an objective standpoint, what is specific about a product that led to such a review. In this paper, we provide a variety of techniques to process textual comments with the goal of rating a product on a numerical scale from one to five. The hypothetical rating is then compared to the actual product rating with the purpose of evaluating the machine learning model. In the future, the team aims to use the techniques discussed in the paper to distinguish between real and fake reviews left by customers to provide an objective way of choosing the best product.**

I. Introduction

The nature of human interaction is extremely complex with a wide range of variation from person to person. Written communication through the English language is such an important aspect of day-to-day life. With the rapid growth of e-commerce and the ever-increasing reliance on the internet, the sheer volume of text available online is incomprehensible. One of the many unique value offerings of e-commerce giant Amazon is its utilization of user-generated content in the form of product reviews. With a star rating system ranging on a scale from one (worse) to five (best), these reviews are essential for customers to better understand the product and to provide unique user feedback.

The mission of the team's analysis was to model the sentiment of millions of Amazon product reviews on various products to predict and summarize the reviews sentiment via the output of a five-class probabilistic classifier.

The team implemented both a random forest model and as well as an artificial neural network to provide probabilistic sentiment classifications on the interval [1, 5]. After the analysis, the team identified the trends within each algorithm and determined the optimal model.

Our dataset was derived from an Amazon review dataset originally from the Stanford Network Analysis Project (SNAP) (http://snap.stanford.edu/data/web-Amazon.html/). It contains 34,686,770 amazon.com reviews from the 18 year period, between June 1995 and March 2013. Reviews included product, user information, ratings, and a plaintext review. The derivation done for the dataset we are utilizing reduced the amount of data operated on and proportioned the data in a way that would represent the five different classifications of our project's goal in equal measures.

II. Related Work

In the research paper Predicting the Semantic Orientation of Adjectives by Vasileios Hatzivassiloglou and Kathleen R. McKeown, the researchers identified and validated the positive or negative semantic orientation of conjoined adjectives within a large collection of written texts known as a corpus. [1]

The semantic orientation or polarity of a word indicates the direction the word deviates from its meaning for its semantic group or lexical field. Within lexical fields, words are grouped by topic and convey a similar meaning to an idea. For instance, the lexical field for the word hiker may consist of the words backpacker, explorer, climber, alpinist, or adventurer. Semantical groups are words that are grouped by meaning. For example, within the semantic field color the words red, blue, yellow, black all share a similar meaning in that they are used to describe color. Semantically similar words are readily identified on the basis of linguistic cues.

Within their research, Hatzivassiloglou and McKeown presented a method that automatically retrieves semantic orientation information using indirect information collected from a large collection of text. The method extracts domain-dependent information from the corpus and adapts to a new domain when the corpus is changed. The method achieves high precision in classifying adjectives and the approach is as follows: Conjunctions of adjectives are extracted from corpus along with relevant contextual information A log-linear regression model combines information from different conjunctions to determine if two conjoined adjectives are of the same or different orientation. A clustering algorithm separates the adjectives into two subsets of different orientations and places as many words of the same orientation into the same subset. The average frequencies in each group are compared and the group with the higher frequency is labeled as positive.

In analyzing opinion-based product reviews on e-commerce websites, researchers working in the Department of Computer Science at the University of Illinois at Chicago mined opinion features on products reviewers have commented on. [2]Their work is related to two areas of research, text summarization, and terminology identification. A majority of existing work on text summarization focuses on a single document covering similar information with the purpose of summarizing similarities and differences in the information contents of the documents. Within the realm of terminology identification, statistical approaches are used to exploit the relationship by which terms are found within other words within a phrase. For example, in the English language, the words brick and mortar might be found in close proximity or are correlated with one another in various documents across the internet. In the work by Minqing Hu and Bing Lui, they studied feature-based opinion summarization of customer reviews in two steps: By identifying the features of the product that customers have expressed opinions on (opinion features) and ranking the features according to their frequencies that they appear in the reviews. For each feature, the researchers identified how many customer reviews had positive or negative opinions. Specific reviews that express these opinions are attached to the feature.

Additionally, in order to discover features not explicitly mentioned in the reviews, it is necessary to implement a semantic understanding of sentences into the model. For example, in the sentences,

”The pictures are very clear”, and ”Overall a fantastic very compact camera”, the product features are not easily discernible and will require more sophisticated techniques to pick out implicit features.

In the paper Character-level Convolutional Networks for Text Classification researchers Ziang Zhang, Junbo Zhao, and Yann LeCun at New York University explored the use of character-level convolutional networks for text classification. [3]

The researchers used a 1-D version of the temporal-max pooling used in computer vision to train convolutional networks deeper than 6 layers. A rectifier was introduced to add a non-linearity component in the model which made the convolutional layers similar to rectified units(ReLUs). The characters in the text had to be quantized using one-hot encoding before it was used as input for the model. Weights within the convolutional network were initialized by using a Gaussian distribution.

To control the generalization error for deep learning models, data augmentation using an English Thesaurus was introduced. Every synonym to a word or phrase was ranked by the semantic closeness to the most frequently seen meaning. The number of words that were replaced was determined by using a geometric distribution. After the model was trained, other traditional and deep learning methods were used to quantitatively evaluate the convolutional neural network model.

Our approach is similar to Hatzivassiloglou and McKeown [1] in that we are retrieving semantic orientation of adjectives, but are not limiting ourselves to one specific word class, i.e. choose to classify adjectives vs. nouns. Our purpose in determining the semantic orientation of word classes is to accurately tokenize words that will be used in the sentence analysis of product reviews.

Within the paper ”Mining Opinion Features in Customer Reviews”, Minqing Hu and Bing Liu focused on mining implicit or explicit product features left in the comments by reviewers. They then tabulated the frequency of positive and negative reviews on each feature to provide future customers with a concise way of evaluating a product. Our team will be mining for product features in reviews to classify them within a five-star ranking system and then use these rankings to predict the actual rating of the product. This way lays the groundwork for distinguishing between real and fake reviews left by customers on a given product.

III. Our Solution

A. Description of Dataset

The derived dataset is constructed by randomly taking 600,000 training samples and 130,000 testing samples for each review score from 1 to 5. In total there are 3,000,000 training samples and 650,000 testing samples. For continuity, the data is split 82.2% for training, 17.8% for testing. The derived dataset can be found at Registry of Open Data on AWS (https://registry.opendata.aws/fast-ai-nlp/), this derived dataset was originally developed for “Character-level Convolutional Networks for Text Classification” by Xiang Zhang, Junbo Zhao, Yann LeCun.

Our dataset is formatted in .CSV format and as reported in the introduction, only 3 of the full datasets features were most relevant to our goal: class index (star rating) [1 to 5], review title, and review text, the rest of the features were trimmed.

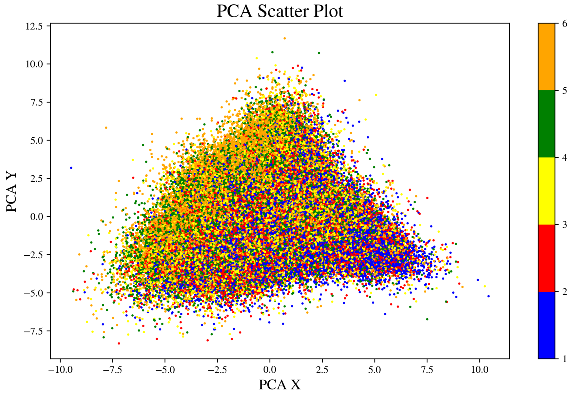
Given that most machine learning algorithms require numerical vector input, the training and testing data must be converted from raw text data to vector form. The team filtered out symbols, stop words, punctuation and converted all characters to lowercase through a process called sentence tokenization. Next, the team removed duplicate or non English reviews. Next, the tokenization process is utilized to maintain the sentiment of the text corpus. After tokenization pre-processing, the tokenized data is normalized using lemmatization. This allows for the full word to be normalized rather than simplified to the root or stem word. Next, the tokenized and normalized data is transformed into a vector through the utilization of the Gensim via Word2Vec [4]. This vectorized output is stored with each variable representing a value on [-1, 1].

For more information pertaining to the original dataset, please refer to the following paper: J. McAuley and J. Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. RecSys, 2013.

B. Machine Learning Algorithms

Upon the vectorization of the raw text data, an optimal machine learning algorithm must be identified and selected. The team will analyze the effectiveness of both traditional machine learning alongside deep learning algorithms. Utilizing ensemble learning through the implementation of a random forest model and an artificial neural network (ANN) multi layer perceptron (MLP), the team will determine the most optimal classification model. Since the vectorized data exists entirely on [-1, 1], there is no need for further data pre-processing or normalization for the MLP implementation. One of the key preferences that impacted the algorithm selection process was the desire to output a probability distribution for each of the five classes. The teams need to output probability distributions for each class rather than just classification predictions that come from the nature of the dataset and the classification features. Due to the fact that sentiment ratings classes are not entirely independent of each other, it is favorable to include classification probabilities in contrast to an individual classification output. For example, a prediction output of a review with strong positive sentiment with higher percentages of ratings belonging to class 4 and class 5 is more useful than a single classification of class 4.

Given the nontraditional format of the vectorized input data and the high dimensions, a dimensionality reduction technique was applied to aid with the model selection process. After the vectorized data was decomposed into two dimensions using the principal component analysis (PCA) technique, the data was plotted in two dimensions to evaluate linear separability. The data was not linearly separable as determined by the PCA scatter plot. This enabled the team to narrow down the model selection process to omit certain algorithms.



From here, it was determined that decision trees could be used to accurately model the vectorized data. Because decision trees are prone to overfitting easily, the random forest method was introduced to leverage the strengths of ensemble learning.

Because the Amazon reviews dataset utilizes training data with millions of samples, the team wanted to also reap the performance benefits of neural networks. Additionally, with deep learning implementations, there is less emphasis on performance attributed to feature selection and more emphasis on direct model implementation. MLPs are a type of artificial neural network and are very useful within the field of linguistic processing. MLPs are often an optimal solution to NLP problems because of the natural approach to capturing the importance of sequential data found within raw text. This is directly applicable to the team's dataset because of the large amounts of raw text data required for analysis. One potential alternative to the MLP model was the recurrent neural network (RNN) or the convolutional neural network (CNN) model. The team will compare the results from the multinomial logistic regression and the MLP to identify the best algorithm.

C. Implementation Details

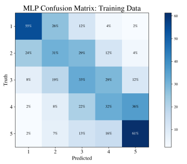
The Scikit Learn Ensemble library provides an off-the-shelf random forest classifier that was implemented to define the classification procedures. This model was configured with the estimators parameter of 50 and the optimization criterion of entropy. Once the model was fit on the training set, the team also introduced a cross validation technique to help better understand the model's performance on various data samples. This was performed with the help of Scikit Learn's cross validation score library with a cross validation size of 3. With the 3 cross validation samples, the differences between the accuracy and F-1 score were negligible. This reveals that the model is not greatly overfit towards the training data and its performance is consistent on additional data samples.

In order to implement the ANN MLP model, the team utilized the Keras library from TensorFlow. A sequential model was implemented with one input layer, four dense layers, and one output layer. The dimensions of the output layer correspond to the probabilistic classification distribution for each sentiment rating 1 through 5. The dimensions of the input layer correspond to the dimensions of the vectorized tokens from the Gensim Word2Vec library.

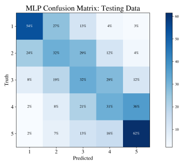
Given the unique architecture of ANN's, another issue that needs to be addressed is the vanishing gradient problem arising from the back-propagation algorithm. Because traditional hyperbolic or sigmoid activation functions such as softmax have gradient derivatives that evaluate very close to zero towards the limits, this vanishing gradient problem becomes apparent with these activation functions. To avoid this issue, the Rectified Linear Unit (ReLU) activation function was utilized as the activation for three of the dense layers with the softmax function used only for the second to last layer and the 5-class output layer. Using the sparse categorical cross-entropy loss function, the model was optimized using the Adam optimizer with a learning rate of 0.01. The number of epochs or weight updates was determined to be 50 because it was evident that the model was converging within the interval of 50 iterations.

There were several measures introduced to combat model overfitting. A kernel regularizer was implemented to penalize values straying too far from the desired target. An L2 or ridge regularization coefficient was introduced as lambda with a default value of 0.01. After loss and accuracy plots were used to compare the convergence rates, it was determined that the final value for the lambda regularization coefficient was 0.001. This is generally smaller than traditional lambda regularization coefficients, but this is because the MLP model didn't overfit that strongly. This enabled the regularization term to be lower than normal.

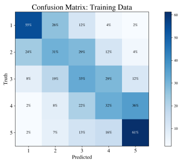
When evaluating the performance of the models, several metrics will help understand model performance. For performance metrics, the team will calculate precision, recall, and F-1 score for each of the classes. Precision will enable the team to determine the ratio of correct class predictions to total class predictions. This will help to identify false-positive predictions. Recall enables the team to determine the ratio of correctly predicted positive classes to every observation in that class to detect false negative-predictions. The F-1 score is used to summarize both precision and recall. Because precision and recall are natively implemented in binary classification problems, there will be some slight modification to implement a multiclass confusion matrix. The multiclass confusion matrix provides additional insight into model performance, as it addresses elements of the model's classification that the F-1 score cannot. Since the desired MLP output is a probabilistic class distribution, the confusion matrix provides better detail into the performance of the MLP. On the other hand, the confusion matrix is not as advantageous with respect to the random forest model, because of the single class output. This explains the differences between the training and testing confusion matrices for the random forest and MLP implementation.



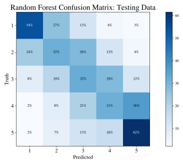
(a) MLP Confusion Matrix: Training Data



(b) MLP Confusion Matrix: Testing Data



(c) Random Forest Confusion Matrix: Training Data



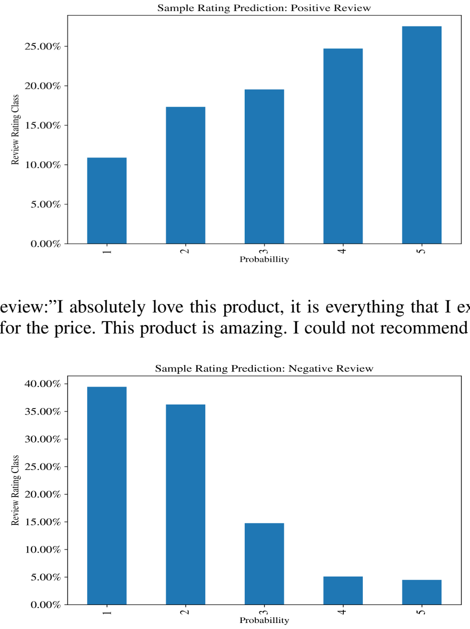
(d) Random Forest Confusion Matrix: Testing Data

IV. Comparison

The random forest model performed with accuracy scores on the training and validation set of 99.99% and 41.28%, respectively. The MLP implementation performed with scores of 42.29% and 42.02% on the training and validation data sets. One primary reason that the training set performed so strongly on the random forest model is that it shows lots of overfitting towards the training data. When performing three cross-validation assessments on the accuracy of the random forest model, the accuracy score on unseen data is always approximately 41%. One fundamental difference between the two implementations is that the random forest model does not output a probability class distribution whereas the MLP neural network model does. Given the probabilistic output, the obtained validation accuracy is favorable since a multiclass probability output provides more information than just a single class output. For example, it is more beneficial to know that a review has high class distributions of strong sentiment (i.e. 4 and 5 star ratings) rather than just knowing the single class of the highest output.

Although the performance metrics for the random forest and neural network MLP models were quite similar, the MLP implementation is much more realistic for a sentiment analysis application. Because of the probabilistic distribution output, the MLP classifier is much more valuable to users because of the ability to output a range of values rather than a single value corresponding to the highest probability.

It should also be noted that the model accuracy of 42% is not extremely high for some machine learning standards, but because of



(a) Positive Review:"I absolutely love this product, it is everything that I expected. The quality is unbeatable for the price. This product is amazing. I could not recommend this product more!"

(b) Negative Review:"The package arrived damaged and the tv was fully broken. Do not buy this product, it is cheap and not made well. The tv is not very bright and makes a loud noise when you turn it on. The screen arrived broken and three days late"

the probabilistic nature of the output and the high overlap between neighboring class features, this result is expected. Because the task at hand is predicting sentiment, the difference between a review with strong and very strong sentiment is not as pronounced as the difference between a dataset with much more defined class features (i.e. benign vs cancerous).

Many existing solutions for sentiment analysis utilize deep learning techniques such as RNNs and CNNs. Because the neural network model that was chosen for this project was a simple sequential model, the analysis was unable to gain the full benefits of deep learning. This was partially due to the high computational complexity and project time constraints. Another solution that is popular within sentiment analysis is the inclusion of an LSTM embedding layer. This is a useful technique combined with RNNs to provide sequential model memory for text processing.

V. Future Directions

If the team was given additional time to advance the project they would implement additional features to increase the endpoint accuracy of the model. Firstly, if given more time the team would seek out additional computing power as it was a bottleneck for the project scope. With additional computing power, the team could have either used a larger portion of the original dataset or increased the complexity of the model. In either or both cases, it would have required more time for computation. To expand on increasing complexity; processing and operating on the Review Title feature of the dataset could allow the team to extract a higher accuracy viewpoint of the user's sentiment. Additionally, the team could increase the number of MLP layers with an end goal of increasing the model's accuracy. If that did not turn out to be as expected, testing other network architectures like: RNN and LSTM embedding, would be the team's next approach. Since the team used MLP as well as an ensemble learning method, Random Forest, the team would look into different ensemble learning techniques as combining results of multiple models generally ekes out additional accuracy; methods like bagging and boosting would be further investigated.

VI. Conclusion

Since the team's mission was to devise a machine learning model to analyze the sentiment of Amazon product reviews, the sentiment of the team was that that mission was successful in the ways that are meaningful. To expand, even though the model the team produced has an accuracy of 42%, as mentioned above, due to the probabilistic nature of the model and the strong overlap between neighboring class, the accuracy of the middle classes (star ratings two through four) have the least accuracy due to having additional neighbors as opposed to the extremities of the rating scale. This is not a major issue however, as observed in similar implementations, thanks to the output of our model being probabilistic, the team could collapse the results down to three classes. This could be achieved by adding an additional layer to the model, those classes would be: negative, neutral, and positive. This would provide a more simplistic but more confident assessment of the user's sentiment, and could ultimately assist fields looking to recommend a user, products based on their sentiment of past products.

References

[1] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*. Madrid, Spain: Association for Computational Linguistics, Jul. 1997, pp. 174–181. [Online]. Available: https://www.aclweb.org/anthology/P97-1023

[2] M. Hu and B. Liu, "Mining opinion features in customer reviews," 07 2004.

[3] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," 2016.

[4] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.