AMAZON FOOD REVIEW CLASSIFICATION USING DEEP LEARNING

Mentors:

Dr. Arpit Bhardwaj

Team Members:

Harshit Singhai
Rishabh Mishra
Yashwant Kotipalli
Rupayan Das
Deepanshu Sehgal
Devansh Pushkarna

INTRODUCTION

Motivation

As the marketplace for consumer products moves to the Internet, the shopping experience changes in a way that makes much of the information regarding the use products available online and generated by users. This contrasts the way that product information used to be disseminated: through word of mouth and advertising. People all over the world are expressing their views, emotions whether positive or negative at some form of social networking platform.

The biggest ecommerce player right now, Amazon, is going through massive growth. Amazon is expected to capture nearly half of the U.S. e-commerce market by the end of 2018; Amazon India is currently worth $16 billion and has the same 30% marketshare as local competitor Flipkart. This makes Amazon an extremely important entity, with lots of relevant consumer data.

Since its creation as an online bookstore in 1994, Amazon.com has grown rapidly and been a microcosm for user-supplied reviews. Soon, Amazon opened its reviews to consumers, and eventually allowed any user to post a review for any one of the millions of products on the site. Amazon has always maintained high emphasis on customer satisfaction. Amazon customer reviews about the products are one of the main reasons to attract customers on Amazon. It basically helps them understand almost every detail of the product. Since, consumers cannot physically inspect the product while shopping online, Amazon product review is the one they can trust in order to judge a product. 90% of consumers online believe their purchasing choices are influenced by product reviews. Many customers consider positive reviews a prerogative to purchasing items online, especially those that cost more. Therefore, it is very important for new sellers to understand that Amazon customer reviews can make or break their e-commerce careers.

It turns out, 90% of the consumers read online reviews before visiting a business and 88% of consumers trust online reviews as much as personal recommendations; customers are likely to spend 31% more on a business with “excellent” reviews; 72% of consumers will take action only after reading a positive review; 72% say that positive reviews make them trust a local business more. These facts serve the purpose to solidify the importance of online customer reviews and opinions.

With this increase in anonymous user-generated content, efforts must be made to understand the information in the correct context, and develop methods to determine the intent of the author. Understanding what online users think of its content can help a company market its product as well as manage its online reputation.

The purpose of this paper is to investigate a small part of this large problem: positive and negative attitudes towards products. Sentiment analysis attempts to determine which features of text are indicative of its context (positive, negative, objective, subjective, etc.) and build systems to take advantage of these features. The problem of classifying text as positive or negative is not the whole problem in and of itself, but it offers a simple enough premise to build upon further.

Problem Statement

With the explosive growth of social media (e.g., reviews, forum discussions, blogs, microblogs, Twitter, comments, and postings in social network sites) on the Web, individuals and organizations are increasingly using the content in these media for decision making. Nowadays, if one wants to buy a consumer product, one is no longer limited to asking one's friends and family for opinions because there are many user reviews and discussions in public forums on the Web about the product. For an organization, it may no longer be necessary to conduct surveys, opinion polls, and focus groups in order to gather public opinions because there is an abundance of such information publicly available. However, finding and monitoring opinion sites on the Web and distilling the information contained in them remains a formidable task because of the proliferation of diverse sites. Each site typically contains a huge volume of opinion text that is not always easily deciphered in long blogs and forum postings. The average human reader will have difficulty identifying relevant sites and extracting and summarizing the opinions in them. Automated Sentiment Analysis Systems are thus needed.
Sentiment is an attitude, thought, or judgment prompted by feeling.

Sentiment analysis (Kim S-M, Hovy E, 2004; Liu B , 2010; Liu B, Hu M, Cheng J 2005; Pak A, Paroubek P 2010; Pang B, Lee L 2004; Pang B, Lee L 2008; Turney PD 2002; Whitelaw C, Garg N, Argamon S, 2005) which is also known as opinion mining, studies people's sentiments towards certain entities. Internet is a resourceful place with respect to sentiment information. From a user's perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites.

Although linguistics and natural language processing (NLP) have a long history, little research had been done about people's opinions and sentiments before the year 2000. Since then, the field has become a very active research area. There are several reasons for this. First, it has a wide arrange of applications, almost in every domain. The industry surrounding sentiment analysis has also flourished due to the proliferation of commercial applications. This provides a strong motivation for research. Second, it offers many challenging research problems, which had never been studied before. Third, for the first time in human history, we now have a huge volume of opinionated data in the social media on the Web. Without this data, a lot of research would not have been possible. Not surprisingly, the inception and the rapid growth of sentiment analysis coincide with those of the social media. In fact, sentiment analysis is now right at the center of the social media research. Hence, research in sentiment analysis not only has an important impact on NLP, but may also have a profound impact on management sciences, political science, economics, and social sciences as they are all affected by people's opinions. Although the sentiment analysis research mainly started from early 2000, there were some earlier work on interpretation of metaphors, sentiment adjectives, subjectivity, viewpoints, and affects (Hatzivassiloglou and McKeown, 1997; Hearst, 1992; Wiebe, 1990; Wiebe, 1994; Wiebe, Bruce and O'Hara, 1999).

However, those types of online data have several flaws that potentially hinder the process of sentiment analysis. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. For example, instead of sharing topic-related opinions, online spammers post spam on forums. Some spam are meaningless at all, while others have irrelevant opinions also known as fake opinions (Liu B, 2014; Jindal N, Liu B 2008; Mukherjee A, Liu B, Glance N, 2012). The second flaw is that ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral.

In this paper we are going to tackle supervised learning problem where we need to predict the positive or negative target variable for each review. The goal will be to maximize the accuracy of this classification. We will train our model on a dataset containing thousands of reviews presented as unstructured text. Each review will be labelled as positive or negative.

Proposed Solution

Most e-commerce websites like Amazon, allow users to leave review about products and services they receive during their shopping experience. These reviews are important to other users when making the decision whether or not to buy the product. Thus, understanding the meaning of the reviews and correctly classify whether it is positive or negative would be a useful thing for the websites to do. Also, the results of the classification can be used for further application such as summarization and recommender system. In this project, we intended to classify the each review as positive or negative using deep learning models. This task can be broken down into three parts: generating a vector representation for each sentence we encounter, using those vectors to train our model, and evaluating the performance of our classification model.

Sentiment Analysis is an important task in NLP. Its purpose is to extract a single score from text, which makes it more convenient to analyze a large corpus of text. Various methods has been used to solve sentiment analysis problems, including bag-of-words and n-grams, and the arrival of deep learning, especially recursive neural network, provides a novel and powerful way to extract sentiment from text data.

Several algorithms like Naive Bayes, Support Vector Machine and Maximum Entropy are commonly used for sentiment analysis. However with the recent development of recurrent neural networks, Deep Learning has started to outperform all the other methods due to its ability to build a representation of whole sentences based on the sentence structure. Several amazing research papers have already been published on this topic by big organizations like OpenAI, Stanford, etc. In this project, we will perform sentiment analysis on the Amazon Fine Food Review dataset from Kaggle.

We will predict if the Amazon Fine Food Reviews are positive or negative using a recurrent neural network. This is a supervised learning problem where we need to predict the positive or negative target variable for each review. The goals will be to maximize the accuracy of this classification. We will train our model on a dataset containing thousands of reviews presented as unstructured text. Each review will be labeled as positive or negative. This paper tackles a fundamental problem of sentiment analysis, namely sentiment polarity categorization (Bhayani R, Huang L, 2009).

RELATED WORK

Work related to Sentiment Analysis

Sentiment Analysis of Customer Product Reviews Using Machine Learning (Zeenia Singla, Sukhchandan Randhawa and Sushma Jain, 2017) In the proposed work, over 4,000,00 reviews have been classified into positive and negative sentiments using Sentiment Analysis. Out of the various classification models, Naïve Bayes, Support Vector Machine (SVM) and Decision Tree have been employed for classification of reviews. The evaluation of models is done using 10 Fold Cross Validation.

Sentiment Classification of Food Reviews (Hua Feng and Ruixi Lin, 2010) presented different neural network approaches including 2 versions of RNN and GRU for sentiment classification on Amazon Fine Food Reviews dataset and reach 68.75% test accuracy in 4 class classification task and 51.74% in 5 class classification task on the test set.

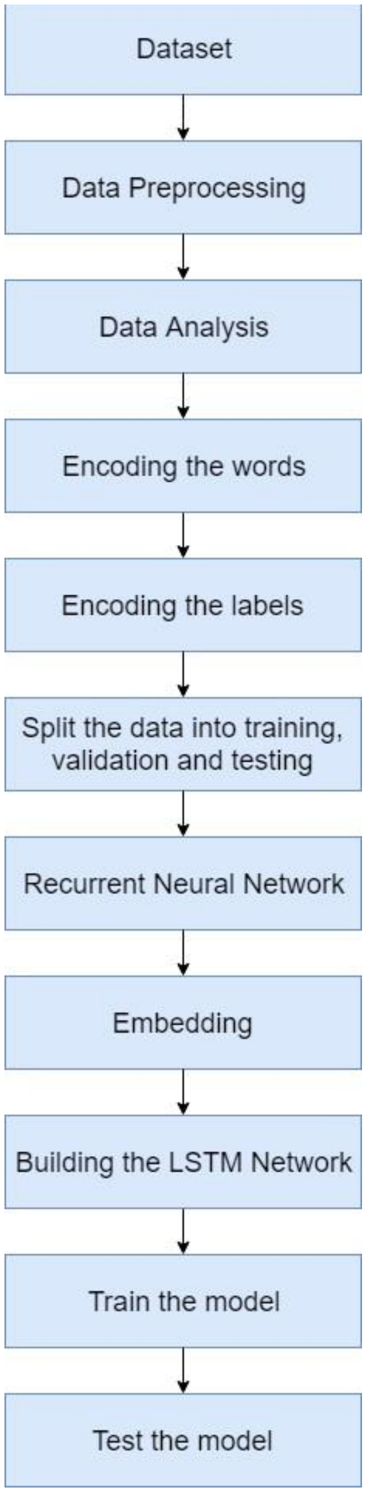
Deep Learning for Amazon Food Review Sentiment Analysis (Jiayu Wu and Tianshu Ji, 2013) the accuracy of their model was 59.32% to 63.71%, depending on different Recursive Neural Network models. They developed vanilla one hidden layer, two hidden layer recursive neural networks and RNTN. To process the raw text data from Amazon Fine Food Reviews, they proposed and implement a technique to parse binary trees using Stanford NLP Parser. In addition, they also propose a novel technique to label tree nodes in order to achieve the level of supervision that RNN requires, in the context of the lack of labeling in the original dataset. Finally, proposed a new model RNNMS (Recursive Neural Network for Multiple Sentences), and have better results than their baseline in terms of every metrics they considered.

Sentiment Analysis of Online Reviews Using Bag-of-Words and LSTM Approaches (James Barry, 2010) performed sentiment classification via two approaches: firstly, a non-neural bag-of-words approach using Multinomial Naive Bayes and Support Vector Machine classifiers. Secondly, a Long Short-Term Memory (LSTM) Recurrent Neural Network was used.

Sentimental Analysis of Amazon Reviews Using Naïve Bayes on Laptop Products with MongoDB and R (Mohan Kamal Hassan, R Sasikala and Sana Prasanth Shakthi, 2017) This paper presents an empirical study of efficacy of classifying product review by tagging the keyword. In the present study, we tend to analyse the fundamentals of determining, positive and negative approach towards the product. Thus this paper hereby proposes completely different approaches by removing the unstructured data and then classifying comments employing Naive Bayes algorithm.

METHODOLOGY

Implementation Design



Dataset

The dataset is made up of 567,363 food reviews that Amazon users had left from November 1999 to November 2012. A review is made up of product and user information, ratings, and a plaintext review. The data extend over a timeframe of more than 10 years, including all 510,010 reviews up to November 2012. After completing the pre-processing of the data, we will be utilising almost 300,000 records of review to help construct our model. Following are the attributes of our data:

Table 1: Attributes of Amazon food review data

Name	**Description**
ProductId	Unique identifier for the product
UserId	Unique identifier for the user
HelpfulnessNumerator	Number of users who found the review helpful
HelpfulnessDenominator	Number of users who indicated whether they found the review helpful
Score	Rating between 1 and 5
Time	Timestamp for the review
Summary	Brief summary of the review
Text	Text of the review

Data Pre-Processing

In order to train our model, we had to transform the reviews into the right format. We performed the following steps:

- Remove punctuations.
- Transform all the characters into lower case.
- Transform each review into a list of integers:
    - First create a dictionary to map each word contained in vocabulary of the reviews to an integer.
    - Then transform each review into a list of integers using the dictionary previously created.
- Create an array with 200 columns, and enough rows to fit one review per row.
- Then store each review consisting of integers in the array. If the number of words in the review is less than 200 words, we can pad the list with extra zeros. If the number of words in the review are more than 200 words then we can limit the review to the first 200 words.

If we don't limit the review to 200 words, it will take too much time for the RNN to train. Here is the distribution of the review length:

The mean and the third quartile of the review length is equal to 79 and 97, respectively. Therefore if we limit the review length to 200 words, we shouldn't lose too much information.

After we binarizing review scores, we notice the dataset is quite imbalanced, i.e. about 80% of the reviews are positive. If we have a classifier which would always predict a review as positive, then it is able to easily achieve 80% accuracy. To solve this problem, we use undersampling technique. We randomly drop positive reviews to make the number of positive reviews are roughly the same as that of negative reviews.

Data Analysis

It is necessary to examine the qualities of our dataset before we begin constructing our deep learning models, to ensure that we are constructing an appropriate model.

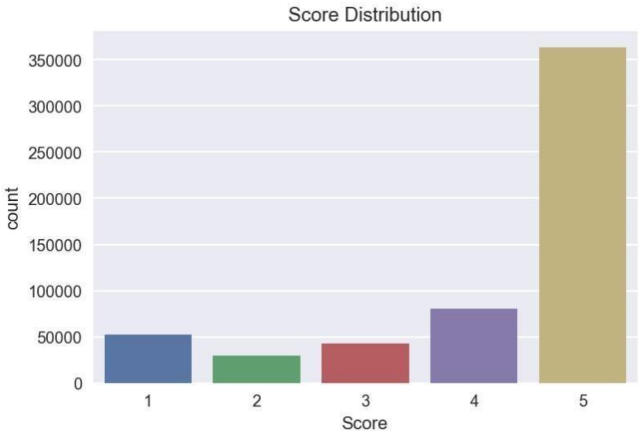


Figure a: Score Distribution Plot

It can be easily observed by looking at the Score Distribution Plot (Figure a) scores 4 and 5 are in majority, with an overall average score of 4.18. This leaning towards one side makes the data distribution skewed towards the left, we will therefore have to make a binary prediction. Therefore, we'll take a score between 1 and 3 as negative and a score between 4 and 5 as positive. Using this binary prediction we'll get two new variables, POSITIVE and NEGATIVE, instead of 1, 2, 3, 4 and 5. We'll name these two new variables as sentiments, the sentiment will be NEGATIVE if the score is equal to 1,2 or 3 and POSITIVE if the score is equal to 4 or 5. Following this convention, we get the following distribution (Figure b):

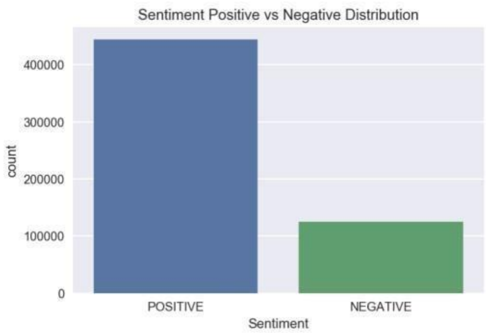


Figure b: Positive vs Negative Sentiment Distribution Plot

Encoding the Words

To map vocabulary to integers, we'll transform the words into integers to create a dictionary for the said mapping.

Encoding the Labels

Range of scores in our dataset ranges between one and five. When calculating the sentiment of a review, a score if greater than or equal to four, the review is considered positive and if score is less than 4, it is considered negative. There to feed our integers in our neural network, score between one and three will be converted into a score of 0, and the score between four and five will converted into a score of 1.

Split the Data into Training, Validation and Testing

The data is split into training, validation and testing set for our RNN model.
Training of this model will be done on training set, validation of the result will be done using validation set. At last, measuring of the model will done by testing it on the testing dataset.

Constructing Recurrent Neural Network

In this project, we use task of predicting positive or negative target variable for each review to evaluate performance of our neural network model. The Recurrent Neural Network is capable of learning extremely complicated relationships from a sequence of data and therefore is considered as being a very expressive model. What makes RNN extremely deep and effective is the fact that it maintains a vector of activation units for each time step in the sequence of data. Two problems arise due to the depth of RNN: the exploding gradient problem and the vanishing gradient problem.

An enforcement of hard constraints over the norm of gradient is used to solve the problem of exploding gradient. Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures are usually used to address the problem of vanishing gradient. Re-parameterization of the Recurrent Neural Network is done by both LSTM AND GRU to solve the vanishing gradient problem.

RNN works on the level of depth it possesses, as we keep increasing the number of Hidden Layers to its model, the speed at which the subsequent hidden layers learn keeps on improving. This happens because the more the number of hidden layers present the more complex arbitrary functions that the network is able to learn and therefore perform a better a job at forecasting future results. We'll be using a Recurrent Neural Network model to predict if a given review is positive or negative. Usually when doing sentiment analysis of sentences, most methods analyze each word individually and assign positive points to positive words and negative points to negative words, and finally add all the points. This method is also known as the lexicon approach for sentiment analysis. This lexicon based approach analyses each word individually and does their sequence, which can lead to the loss of essential information. The Recurrent Neural Network model doesn't make this mistake of analyzing texts at face value, it creates abstract representation of its learning and therefore is able to understand the subtleties.

Adam optimizer is used to minimize our cost function in order to make our network output as close as possible to the target values. Adam is an optimization algorithm that can used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

Embedding

In order to symbolize words by vectors we will be using embedding layer. Given the fact that the size of the vocabulary we're using in this project is more than 240,000 words, representation using embedding layer is much more efficient that one-hot encoding. To ensure that words when having similar meaning are close to each other, we'll using the word2vec approach to generate our embedding.

LSTM

To improve our results and explore more advanced models, we adopted Hochreiter's and Schmidhuber's LSTM model (which also addresses RNN's vanishing gradient problem). It is a special case of Recurrent Neural Network. Long Short Term Memory model, as the name suggests has the advantage of being able to memorize information. Instead of using pre-trained word vectors (as we did in case of RNN), we'll train our word vectors on a global word-word co-occurrence matrix (all words of all reviews). We'll insert our representation vectors into our Long Short Term Memory model. A mapping of word sequences to a class using this model has been illustrated in Figure e.

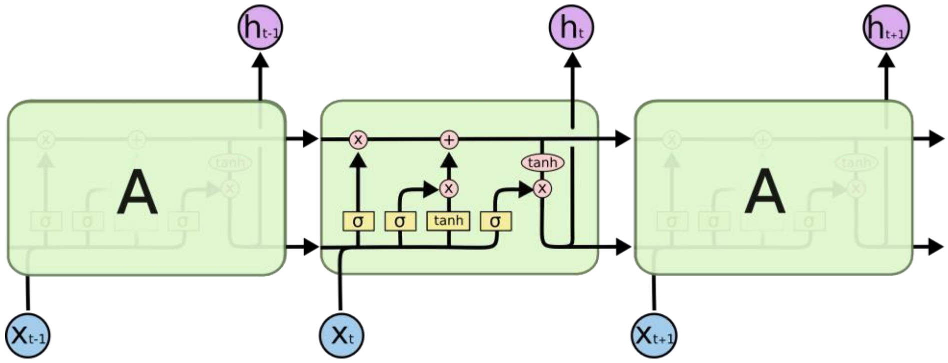


Figure e: Work flow of LSTM From Colah's Blog- Understanding LSTM (colah.github.io/posts/2015-08-Understanding-LSTMs/)

Train the Model

We'll train our model as our next step. Stochastic Gradient Descent method will be utilised. An estimate of the loss function will be calculated (Cross Entropy in this case) on small batches of dataset in this approach. This approach of Stochastic Gradient Descent is more scalable than traditional Gradient Descent.

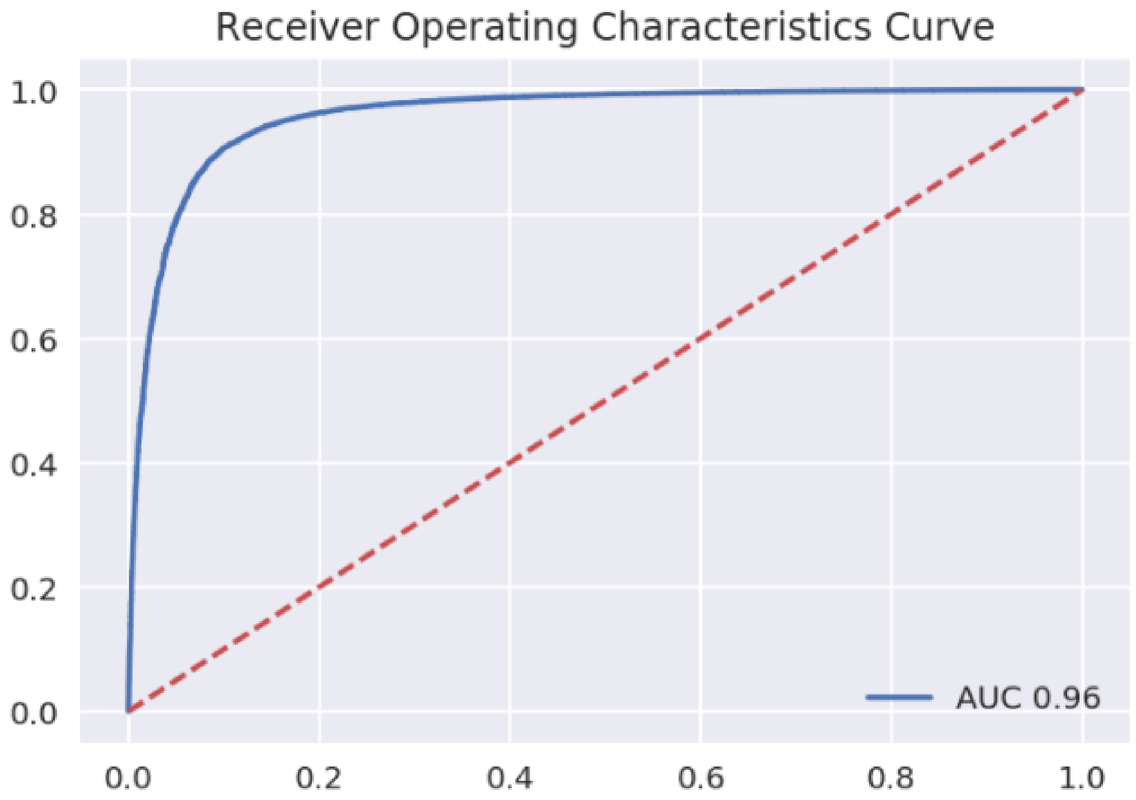
RESULT

Model Evaluation and Validation of LSTM Network

Based on the observation above for our final model, we have decided to include all the vocabulary, have a single LSTM layer, an LSTM size of 256, and a dropout layer with a probability of 0.5.

The best way to see if our model is able to generalize is to test our model on the untouched testing set. In order to measure the performance of the binary classifier, we can plot the ROC curve.

AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represents degree or measure of separability. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. In our case, Higher the AUC, better the model is at distinguishing between reviews which are positive and negative. The ROC curve consists of plotting the true positive rate against the false positive rate. The best scenario is when the curve matches the left corner. This would mean that we are able to achieve 100% sensitivity (no false negatives) and 100% specificity (no false positives).



We can observe that the area under curve (AUC) is equal to 0.96. Knowing that maximum AUC score is equal to 1, we can conclude that our model does a pretty good job at making predictions on unseen reviews. Moreover our model seems to be very balanced in terms of false positive rate vs false negative rate.

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision = \frac{\text{True Positive}}{\text{True Positive } + \text{False Positive}}$$

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class.

$$Recall = \frac{\text{True Positive}}{\text{Total Positive} + \text{False Negative}}$$

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

$$F1 = 2\text{ X } \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

The other important metric is the F1 score, because it also takes into consideration, precision and recall. In our original dataset 78% of the reviews were positive. If we would use a naïve algorithm and predict that all the reviews are positive, our accuracy would be 78%. This is why we should use the F1 score in order to measure the performance of our RNN.

	precision	recall	f1-score	support
0.0	0.87	0.77	0.82	12363
1.0	0.94	0.97	0.95	44137
micro avg	0.93	0.93	0.93	56500
macro avg	0.91	0.87	0.89	56500
weighted avg	0.92	0.93	0.92	56500

('Accuracy ', 0.92564601769911503)

The final F1 score is equal to 0.92. The precision and recall are also equal to 0.92. It means once again that our model has a good balance between sensitivity and specificity. There is room for improvement but our RNN model has learned to distinguish negative fine food reviews from positive reviews.'

If we compare the performance of our RNN and the benchmark, we can observe that they are very similar. However our model was based on the review input variable and the benchmark was based on the summary variable. Even if the reviews were very detailed with vocabulary specific to the products, the RNN is able to identify relevant information to determine if these are positive or negative reviews with an F1 score of 0.92.

With such performance, we could definitely use this algorithm to identify if fine food reviews are positive or negative.

COMPARISONS

S No.	Other Works	Dataset	Method	Classification accuracy (%)	F-1 Score	Our Model	Our F-1 Score
1.	James Barry [2010]	Amazon Fine Food Review dataset	LSTM	94.77%	--	LSTM	0.92
2.	Hua Feng, Ruixi Lin [2010]	Amazon Fine Food Review dataset	RNN	68.75%	--		0.92
3.	Jiayu Wu Tianshu Ji [2013]	Amazon Fine Food Review dataset	RNNMS	63.71%	--		0.92
4.	James Barry [2010]	Amazon Fine Food Review dataset	SVM	87.15%	--		0.92
5.	James Barry [2010]	Amazon Fine Food Review dataset	LSTM	94.77%	--		0.92
6.	Hua Feng, Ruixi Lin [2010]	Amazon Fine Food Review dataset	GRU	51.74%	--		0.92
7.	Jiayu Wu Tianshu Ji [2013]	Amazon Fine Food Review dataset	RNTN	59.32%	--		0.92
8.	James Barry [2010]	Amazon Fine Food Review dataset	Multinomial Naive Bayes	90.42%	--		0.92
9	Joseph Roy[2015]	Amazon Fine Food Review dataset	Decision Tree	83.42%	0.485		0.92
10.	Joseph Roy[2015]	Amazon Fine Food Review dataset	KNN kd-tree	84.347%	0.283 --		0.92

CONCLUSION

In this paper, we were able to demonstrate that we can use RNN to predict sentiments. Once again, the strength of the RNN is that it's able to take into consideration the entire sequence of words, which help the model find nuances. The main challenge of using a deep learning approach is that it's very expensive in terms of computing and time.

We presented neural network approach of LSTM for sentiment classification on Amazon Fine Food Reviews dataset and reached 0.92 F1 score using single layer LSTM of size 256 with drop out probability 0.5. In our experiment, we find that padding zeroes to reviews proves to be useful and the zeropadded approaches outperform the approaches without zero-padding we implement.

To train this network we had to use GPUs provided by the platform FloydHub. FloydHub is a managed cloud platform for data scientists, it provides them tools to make every step of their workflow productive. FloydHub contains fully configured development environments for deep learning on the cloud.

Even with GPUs, it takes at least 1.5 hours to train our model. It was definitely a challenge to build our model and tune the hyperparameters correctly.

FUTURE WORK

Future work might focus on trying out more RNN models, like the bidirectional RNN and tuning other parameters like hidden layer size and number of steps. Our analysis could be a useful tool to help restaurants better understand reviewers' sentiment about food, and can be used for other tasks such as recommender systems

Deep Learning techniques tend to outperform every other algorithm when we get access to lots of data. So if we could train our model on a larger dataset, the model should definitely improve. With RNN, the model creates its own representation of the sentence. The reviews contain vocabulary specific to the food product. If we had mode data, our model should be able to identify the specific characteristics that make a good product. In order to improve this model, we should also investigate if we could include the product name/product type. If we had this information maybe our RNN would be able to more easily identify the important characteristics for each product.

We will perform a thorough error analysis in the test dataset and we will investigate why some sentences have been wrongly classified. We aim to play with the features used in each of the sub-classifiers in order to better understand their respective contribution in the classification result. Moreover, we will experiment with different aggregation functions such as a weighting schema that would give priority to a particular classifier. Last but not least, we will investigate further how to improve the scalability of the Stanford Sentiment System.

REFERENCES

Akkaya, Cem, Janyce Wiebe, and Rada Mihalcea. Subjectivity word sense disambiguation. in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP2009). 2009.

Bhayani R, Huang L (2009) Twitter sentiment classification using distant supervision, 1–12.. CS224N Project Report, Stanford.

Hatzivassiloglou, McKeown. Predicting the semantic orientation of adjectives. in Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-1997). 1997.

Hearst, Marti. Direction-based text interpretation as an information access refinement, in Text-Based Intelligent Systems, P. Jacobs, Editor 1992, Lawrence Erlbaum Associates. p. 257-274.

Kim S-M, Hovy E (2004) Determining the sentiment of opinions In: Proceedings of the 20th international conference on Computational Linguistics, page 1367.. Association for Computational Linguistics, Stroudsburg, PA, USA.

Liu B (2010) Sentiment analysis and subjectivity In: Handbook of Natural Language Processing, Second Edition.. Taylor and Francis Group, Boca.

Liu B, Hu M, Cheng J (2005) Opinion observer: Analyzing and comparing opinions on the web In: Proceedings of the 14th International Conference on World Wide Web, WWW ’05, 342–351.. ACM, New York, NY, USA.

Mohan Kamal Hassan, R Sasikala, Sana Prasanth Shakthi (2017) In: Proceedings of IOP Conference Series: Materials Science and Engineering.

Pak A, Paroubek P (2010) Twitter as a corpus for sentiment analysis and opinion mining In: Proceedings of the Seventh conference on International Language Resources and Evaluation.. European Languages Resources Association, Valletta, Malta.

Pang B, Lee L (2004) A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts In: Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics, ACL ’04.. Association for Computational Linguistics, Stroudsburg, PA, USA.

Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr2(1-2): 1–135.

Sukhchandan Randhawa, Sushma Jain, Zeenia Singla (2017) Sentiment analysis of customer product reviews using machine learning In: Proceedings of International Conference on Intelligent Computing and Control (I2C2).

Turney PD (2002) Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL ’02, 417–424.. Association for Computational Linguistics, Stroudsburg, PA, USA.

Whitelaw C, Garg N, Argamon S (2005) Using appraisal groups for sentiment analysis In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM ’05, 625–631.. ACM, New York, NY, USA.