ESOF 2918 Final Report

Nicholas Imperius: 0645031, Kristopher Poulin: 0883832, Jimmy Tsang: 1098204

Lakehead University

Thunder Bay, Ontario

Supervisor: Dr. Rachid Benlamri

Abstract - In the field of marketing, customer feedback is crucial, but companies tend to not pay enough attention to the customer reviews. The success of a product highly depends on the public opinion, which is sometimes ignored. The main focus of this paper involves using Natural Language Processing in conjunction with the open source software spaCy to analyze sentiment analysis of customer reviews by accurately classifying reviews that enable for a more precise representation of the reviews for a particular product. We conducted a series of tests and displayed our findings in neat, concise tables and graphs. As a result of this project, E-commerce platforms that absorb the responses of their customers using a sentiment analyzer will gain an advantage in developing and producing a more attractive product.

Terminology

Natural Language Processing (NLP):	A subfield in computer science that deals with how to program a computer to process and analyze a large amount of natural language information.
Sentiment Analysis:	A technique used to systematically identify, extract, quantify, and study information to determine whether it is positive, negative or neutral.
Opinion Mining:	A synonym for Sentiment Analysis.
Prototype:	A preliminary model/sample of a product built in order to test a concept or process.
Python:	An interpreted, high-level and general-purpose programming language.
Rating:	A score given to represent the overall feeling of the review (Higher Rating = Better Review, and Vice Versa).
Precision:	Calculated by dividing the number of True Positives by the number of True Positives and False Positives.
Recall:	Calculated by dividing the number of True Positives by the number of True Positives and False Negatives.

Loss: The amount of error in the training process.

I. Table Of Contents

II. Executive Summary

The problem we defined is that online retailers do not take value in customer reviews as much as they should. Lots of retailers do not have a deeper understanding of why their customers like or dislike their products because the main priority is just to get the product out and sell it. In addition, some users do not leave accurate reviews, for example, a review on a product is provided that consists of points as to why the customer is satisfied with the product but because the packaging was damaged, the consumer gave it a negative review.

The importance of discovering a solution to this problem is that it will increase customer satisfaction and review the accuracy, which in turn, leads to bigger growth for a business. For example, if a company develops a product that consumers have a distaste for, they can look at the feedback in reviews in order to pinpoint what areas to improve or change. Afterwards, they can then make changes to the preexisting product in order to satisfy their customers, and with more satisfied customers, there will be more demand for their future products.

Our proposed solution will allow businesses to automate the process of analyzing reviews and understand why customers are leaving those reviews and what changes they want to see in the products. Using sentiment analysis, we can classify each review from one to five stars instead of just looking at it from just a positive or negative standpoint, which gives us more information on the sentiment behind the review.

The rest of this paper is structured as follows. Section III gives a literature review as well as the problem and reasoning behind the work for the literature. Section IV shows our approach to the problem. Section V is what our problem is as well as the benefits of solving the problem. In Section VI we discuss the design and implementation of our prototype. In section VII we look

at the results retrieved from running the prototype. Section VIII is our concluding statement. Finally, Section IX is our self-reflective reports.

III. Technical Background

A. Survey and Critical Appraisal

Online reviews provide a basis for customers to gain an opinion on whether or not a certain product is worth purchasing. In addition, it gives an opportunity to the manufacturer of said products the ability to refine features based on the feedback received from users. As shown by Jin *et al.* [1], analyzing consumer reviews and feedback for choosing products they were able to relay information back to the creators of the products to refine them to be better. [1] also uses frameworks that calculate the effect that certain product features have and by understanding and reviewing this feedback the brands are able to gauge overall customer satisfaction. Chauhan and Sehgal [2] explain how companies do not quite understand fully what the customer needs. In addition, [2] discusses how it is difficult for customers to construct one single review of all reviews and relies on the importance of sentiment analysis which allows customers to be able to choose the best product for them. This is because with more and more shopping happening on e-commerce platforms there are little to no human interactions taking place for other people's opinions and the only source of that is the reviews left by other people.

Upon reviewing many journal entries and conference papers on NLP we have come to realize most studies show there are 3 levels to the structure of opinion mining. For instance, Chitra *et al.* [3] stated the 3 levels to be Collection and Extraction, Processing, and Results. Collection and Extraction refer to the preprocessing techniques and tokenization that are used on most systems. Processing is described equivalently to the analysis of sentiment in the reviews. The results are how we classify and represent the data once it has been analyzed (processed).We

noticed a trend of similar 3 phases in our research; however, we found some more stages depending on the requirements they are trying to fulfil. Also, we found that even though the name they gave to each phase may have been different, the idea behind what is accomplished in that phase remained the same. Solangi *et al.* [4] wrote that there are 4 stages that all deal with preprocessing of the data alone followed by the phase of using programming tools to create results. The first stage consists of tokenization which is described as removing irrelevant wording such as stopwords. Followed by the second stage, segmentation, [4] states that other dialects like Chinese or Japanese do not have a lot of major tokenization compared to English. The third stage dealt with deep learning techniques and word embedding techniques. The final stage consisted of strategies such as Parts of Speech Labeling and Parsing that dealt with the syntax of the sentences. In addition, Lai and Hsu [5] described 3 key techniques, The Extraction Rule, The Dictionary Method, and Rating Prediction based on Text Analysis which we found to be in line with how we plan on solving the problem for our system.

Challenges in the field of sentiment analysis mainly dealt with human errors in constructing the review. For instance, bad grammar and/or spelling mistakes would create discrepancies in the accuracy of the system since it would not have learned those mistakes and how to deal with them. [4] when they mention how some NLP techniques would have to be modified in order to resolve this issue but did not go into detail on what the solutions would look like. Performance of other authors' systems showed results of being effective. In [6], the system is given hotel reviews and designates the review as positive, negative, or neutral. The system was found to be 93% accurate. [6] also broke up the testing into different groups, such as reviews about accommodations, food, service, and general aspects. All categories except general aspects achieved essentially the same accuracy rating while general aspects was 5% higher; leading me

to believe that the more general the wording the more accurate the system was. Moreover, using more technical terms to describe food or service satisfaction creates a higher difficulty in understanding the feeling behind the sentiment.

Sentiment analysis methods that seek to summarize the feeling behind a review proved to be a difficult task [7]. Furthermore, there are many tasks that are required to complete this goal that consistently assumes there are no grammatical or spelling errors which in real life is a common issue. However, systems do prove to be most effective enough to allow us to still get an understanding of the customer's feelings providing benefits to the users of the system.

B. Identification Of The Problem

Given a set of reviews, our system will perform tokenization on each review. To do the tokenization, the system will first preprocess the sentences in the review followed by extracting words into tokens. The tokens will then be preprocessed by removing any stopwords and irrelevant punctuation since they do not provide any benefits in gathering the sentiment of the sentences. We will then use NLP algorithms from the spaCy open source library, which are a form of word vectorization combined with text categorization, to analyze and determine the sentiment in the reviews. The final results will then be compiled into a .csv file for when datasets are tested or a simple rating and score are displayed in the console window for single review use.

C. Justification For Proposed Work

Having a valuable resource that allows you to gather feedback from customers on how well a product is performing is vital. With e-commerce becoming the way for the majority of shopping now, more and more customers are relying on reviews made by previous buyers. This

new era of social media and the new advances made to web technologies are becoming more evident and marketing is becoming even more important. A key aspect of marketing is understanding the target audience and their feelings. Being able to have a tool that can analyze these reviews on customer service, product performance, general feedback, etc. will prove to be vital in the growth of a company or product. For instance, Amazon is a large company that has an e-commerce platform that is widely respected and is dealing with thousands of reviews, possibly per hour. Thus, having a system on hand that could analyze the reviews would give them a marketing advantage over companies that do not have such systems in place.

IV. Technical Approach

A. Objective

The aim of this project is to develop a system that can provide the following services:

- Preprocess reviews
- Tokenize the sentences
- Analyze the sentiment in sentences
- Classify the sentiment

B. Project Tasks

To preprocess the reviews we plan on implementing techniques similar to how Bibi [8] described, that is, removal of stopwords, blank reviews, and reviews in different languages. In order to remove stopwords, we must create a library of predefined words to compare each word to, a similar list will be defined for all punctuation as well.

To tokenize the sentences, the idea is to separate each word and punctuation into a separate token [9]. This will be done by looking through each character and detecting if the previous or following character is punctuation or space. Depending on the type of punctuation being used, the appropriate action will be taken. For instance, a hyphen will signify that the token is not over and to continue to accumulate characters preceding the hyphen while a comma or period represents the end of the word thus creating a token.

Analyzing the sentiment in sentences will be done by detecting the level of positivity or negativity or emotion in a particular token. Once we have trained our system with example data sets, the system will develop rules to compute how positive or negative sentences and tokens are.

After each review is analyzed, our system will then give an overall rating to classify the review with a rating from one to five, with one being very negative and five being very positive. The rating will be given depending on the frequency of key tokens, as well as how positive or negative the tokens are. We will have to train our system to perform proper classification based on these metrics.

C. Project Schedule

The schedule followed throughout the progress of this project:

Week 1-2: Choose a supervisor as well as think of project ideas.

Week 3-4: Choose an idea based on recommendations during our first meeting with our supervisor and submit a completed project proposal.

Week 5-6: Research related literature relevant to our problem, worked on progress report

Week 7: Submit completed progress report

Week 8-9: Construct and present our first oral presentation

Week 10: Begin collecting and organizing the data to be used

Week 11-12: Manually classify reviews and begin coding of our prototype

Week 13: Test and train models and compile data from running our prototype

Week 14: Complete final report and present our final oral presentation

V. Problem Analysis

A. Identification, Formulation, and Analysis

The problem that our project outlines are that consumer satisfaction must play a bigger part in online shopping. Rather than simply looking at a score rating, the whole review itself must be analyzed and their feedback must be taken into consideration to improve the score rating. Our prototype plans to analyze a review, and understand why that rating was given to the product.

The idea for this project came to fruition as two of the group members currently work in retail position jobs, where customer satisfaction with items is very important. Therefore, we would like to develop a program that can be used in this field.

We have determined that Natural Language Processing is an effective method for capturing and analyzing the views of the consumers in order to better understand what they are trying to put across. In spaCy, there are built-in functions that will allow us to create pipelines that enable us to create and train a text categorizer model. We will be using the Word Vectorization method that is built into spaCy to help classify our review. Word Vectorization is a method that maps words or phrases from vocabulary to a corresponding vector of real numbers in order to find word predictions, similarities, and semantics.

B. Anticipated Benefits

We anticipate that this project will not only seek to provide better information on the positivity of a review but also further fill in any gaps in the field of Natural Language

Processing. By solving our problem, we will be able to allow businesses and consumers to get a more accurate rating of a product at and quick glance by just seeing the rating instead of having to read through each review to formulate your own opinion based on their opinion. This has an enormous impact on e-commerce, as businesses who utilize this take into account what their customers are saying about their products are more successful in designing products that consumers enjoy, over those that do not take into account the reviews of customers. Also, with a more accurate representation of what customers like and dislike, companies will have an advantage over others in regard to marketing as they are better able to understand the audience.

VI. Design and Implementation

A. Dataset

For the testing and training of our project we created two data sets that are a small subset of the data set of appliance reviews provided by http://deepyeti.ucsd.edu/jianmo/amazon/index.html which initially contained over 600,000 reviews. We created a python program to go through 600,000 reviews and choose 50,000 random reviews and another 1,800 random reviews and create two separate files for both. The first data set of 1,800 reviews was the data set that we used to manually rate each review, we each were assigned 600 reviews to go through which would combine our biases together. To test and train our project, we put 1,200 randomly selected reviews into a training folder and the remaining 600 into a testing folder. The second data set of 50,000 reviews was separated in half, 25,000 into a training folder and 25,000 into a testing folder. During our manual rating process that we each went through, some difficulties we encountered were reviews in different languages, empty reviews, as well as extremely long reviews with conflicting opinions. For instance, someone would create their review but then two months later, they would update the review to have new information about the product that would either be the same sentiment or the opposite.

B. Design

When running the application, we have the option to either train the model, test a single review, or test a batch of reviews against the trained model. The preprocessing of information comes just before the training process begins, the system will remove any unwanted information

such as stopwords and special characters such as '\n' and '
</br>'. The training process consists of training 20 iterations with each iteration learning from the reviews that were rated. In each iteration, we were given the precision, recall, and F-score as well as the loss. We used a multi-class metric system to construct a confusion matrix which allowed us to compute the precision, recall, and F-score in each iteration. This confusion matrix needed to be used instead of a traditional F-Score metric system that used the terms: True Positive, False Positive, True Negative, False Negative since we now have five possible outcomes instead of two. An example of our confusion matrix is shown in figure 1 below. In this case, the yellow numbers represent false positives, the orange numbers represent false negatives, and the green numbers are true positives. For instance, as shown in figure 1, if the system were to predict a one-star but the actual rating is not a one, this is a false positive. However, if the system were to predict a three-star but the rating was actually one-star, then this would be a false negative. In theory, there are no true negatives since a true negative can be either a true/false positive or a false negative. As soon as there are more than two outcomes, a true negative cannot be distinguished uniquely from a false positive or false negative. For instance, in figure 1, if the system predicted a two-star review to be a one-star, according to our formulas, we treat that as a false negative instead.

		Actual				
		1	2	3	4	5
Predicted	1	58	22	8	9	6
	2	49	35	20	8	6
	3	28	26	31	16	3
	4	14	18	29	28	16
	5	7	13	18	51	81

Figure 1: Confusion Matrix Example

In figure 2 we graphed the loss encountered during one of the training processes in our manually rated model. As you can see, the loss decreases with every training iteration. After the

training process was completed, the model would be saved to your computer files system to be loaded later for testing purposes.

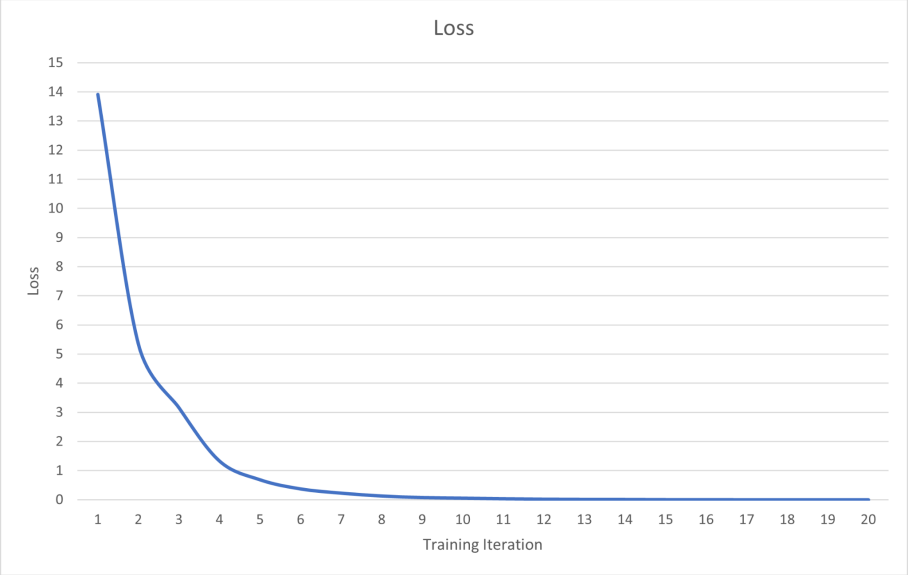

Figure 2: Graph of Loss vs Training Iteration

To test our data sets, the system loads in the .json file and stores the review and rating in an array and stores that array in an array to create a two-dimensional array. The system would go through each review individually and test it with the specified trained model and then compare it to the original rating. A multi-class metric system was implemented here as well so at the end of testing we can calculate the precision, recall, and F-score of that dataset for the particular model as seen in figures 3, 4, and 5 below.

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

Figure 3: Precision Formula

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

Figure 4: Recall Formula

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Figure 5: F-Score Formula

The program saves the review, model rating, original rating and the score given to how sure the model is of the rating to a .csv file and saves another file with the precision, recall and F-score review, this is an accumulation so as the testing continues this value gets more accurate of the final value. You will see this data in Section IV. In addition to testing large data sets, we implemented a feature to allow for single-use so that the user can run the trained model against a single review and the system will output the rating as well as the score associated with it.

C. Implementation

We created the project as a python application that is run on a command line. With the use of spaCy - version 2.3, which is an open-source natural language processing library, we were able to successfully create and train a Text Categorizer model. SpaCy has a large amount of documentation that aided us in designing the system. The training process sorts through five folders that each contain reviews for a single rating with each review being a .txt. The testing process uses a .json file that stores both the review and the rating associated with it. To speed up the testing process we translated all of the .txt reviews that will be used for testing into a .json for quicker runtime, instead of reading 600 or 25,000 individual files we only read one. This sped up testing from 3-4 hours to around 45 minutes for the 25,000 files. In addition, we were able to reduce the amount of code required in our testing methods.

VII. Investigation

A. Description of Appropriate Experiments

We did 6 different tests on our prototype and measured the accuracy of our system by comparing the model rating to the original/manual rating given to the review. In table 1 below, we display the accuracy of each test if both ratings are equal, the accuracy of the model rating is plus or minus one of the original or manual rating, and the F-Score that has been determined.

Test #	**Description**	**Accuracy**	**Accuracy (+/- 1)**	**F-Score**
1	*Trained with:* Manually rated dataset *Tested with:* Manually rated dataset that does **not** contain stopwords	34.17%	71%	0.375
2	*Trained with:* Manually rated dataset *Tested with:* Manually rated dataset that contains stopwords	38.83%	77%	0.329
3	*Trained with:* Manually rated dataset *Tested with:* Original reviewer rated dataset that does **not** contain stopwords	31.67%	69.33%	0.312
4	*Trained with:* Manually rated dataset *Tested with:* Original reviewer rated dataset that	39.67%	77.33%	0.359

	contain stopwords			
5	*Trained with:* Original reviewer rated dataset *Tested with:* Original reviewer rated dataset that does **not** contain stopwords	36%	75.5%	0.358
6	*Trained with:* Original reviewer rated dataset *Tested with:* Original reviewer rated dataset that contains stopwords	37.33%	75.16%	0.393

Table 1: Experiment descriptions and results

As you can see, we completed a test with and without stopwords in each scenario. We trained a total of four models. Two models were trained on the manually classified reviews and the other two were trained on the original rated reviews. One from each pair was trained with stopwords in the training data and the other did not contain stopwords, we did this by removing the line of code that removes the stopwords. We decided to run this test comparison because we noticed some differences in the performance on a single review basis when we tested and trained with or without stopwords and we wanted to compare and see if it was actually something worth doing and possibly think of reasons for the difference.

B. Interpretation of Results

To evaluate the system, we used the F-score metric, which is the harmonic mean of precision and recall, as seen before in figures 3, 4, and 5, and we computed the accuracy, which is the total positive values divided by the total number reviews in that testing set as seen in figure

6. The F-score is calculated based on two merits, precision and recall. For our prototype, we used a multi-class metric system since there are five possible outcomes now.

$$Accuracy = \frac{True\ Positive}{Total\ Reviews}$$

Figure 6: Accuracy Formula

Throughout our experiments, we found that our model had an accuracy rating of 34.17% and an F-score of 0.375 when tested and trained using our manually classified reviews without stopwords.

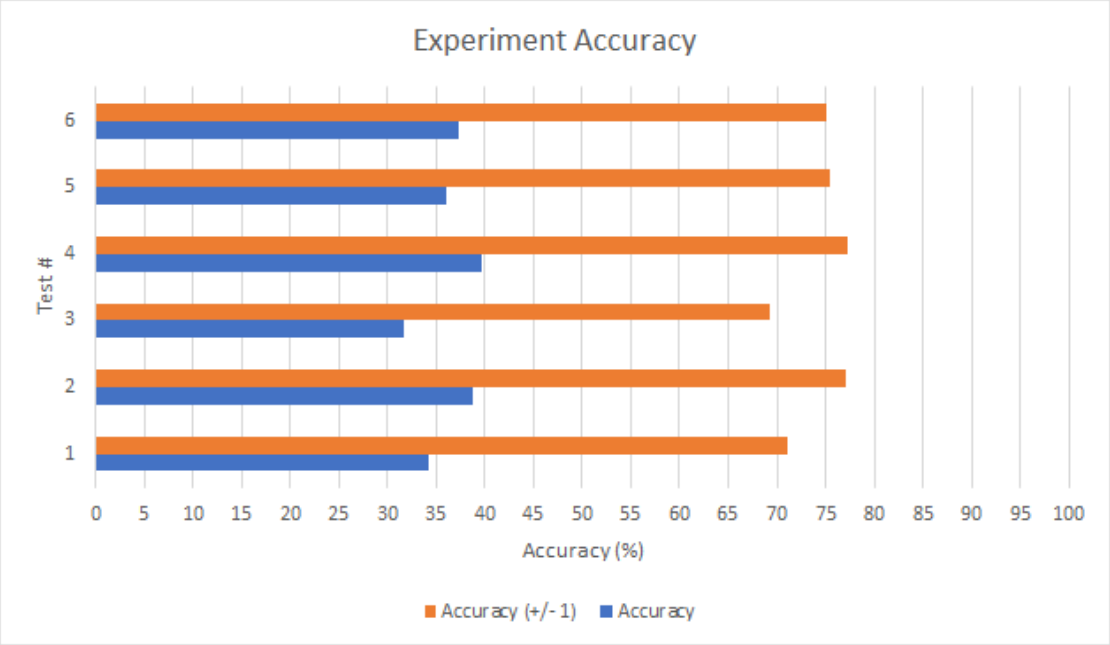

Figure 7: Accuracy for each test

As seen in figure 7, we computed another accuracy rating that allowed a variance of one star, what this means is that if the original rating was a four and our model predicted either a three, four, or five stars, we would consider this correct. There are many small variables that can change a rating from a four-star to a more neutral three-star as well as changing from a four-star

to a more positive five-star based on the sentiment the model discovers. In this scenario, our model now had an accuracy of 71%. Some of the small variables include bad spelling, removal of stopwords in more negative reviews, and sarcastic writing. With bad spelling and improper sentence structure, the model has a hard time identifying stopwords as it looks for a specific word and if it is not correctly spelt it will not be recognized. In addition, with the more negative reviews, words that would normally indicate a negative sentiment are treated as stopwords and thus removed. This includes words such as ‘not’, ‘no’, ‘bad’, and ‘down’, which completely changes how the sentence is read and the model can potentially determine the sentiment to be more positive. Moreover, the model has a troubling time detecting sarcastic writing as it is written to be jokingly positive, which we, like people, can tell this statement is meant to be sarcastic but the model cannot. For example, a one-star rating states, “I love this device, it only stopped working after 1 month”. In this situation, the model would have a hard time determining that the sentiment is actually negative, where we can easily detect that it was a sarcastic statement. Throughout our manual classification process, we observed that many original ratings were very different from the sentiment in the review. For instance, a reviewer would mention how great a product was but give it a two or three-star rating which does not accurately reflect the sentiment in the review. In regards to the increase of accuracy when allowing for a variance of one star this could be due to personal opinion on certain words. For instance, when writing a review, some people might see a review stating ‘Ok’, as a four-star, whereas some people might perceive it as a five since nothing was wrong or a three-star since it is not ‘good’ or ‘great’. We split the number of reviews that we were classified into three equal portions which would be able to give the model when trained, an average opinion bias of all of us.

In the initial development phase of our prototype, we noticed a difference incorrectness of the model when trained and tested with data that still contained the stopwords vs the same data without stopwords. In tests 2, 4, and 6, you will notice that both the accuracy and F-score are higher than the same test without stopwords. Table 2 and Table 3 show our findings when looking further into this. In some instances, key terms that helped with predicting the sentiment were removed and this made it more difficult for the system to give an accurate rating. We can prove this since with each prediction that our model gives, it also provides a score. This score is a representation of how confident it is of the prediction being correct. In test 2, the average score was 0.708 while in test 1 the average score was slightly less at 0.682.

With Stopwords	**Without Stopwords**
Living off grid, this washer worlds great! Does multiple load types and uses limited water and power. The spin feature is nice, so you clothes isnt soaking wet when done and can be hung on the line to dry.	*Living grid , washer works great ! multiple load types uses limited water power . spin feature nice, clothes soaking wet hung line dry .*
Love everything about this machine. Saving up to buy the little dryer!	*Love machine . Saving buy little dryer !*
Easy to use I got mine today.	*Easy use got today .*
Absolutely horrible. It arrived with either a broken pump, or electrical board and was	*Absolutely horrible . arrived broken pump, electrical board totally useable . Shipping*

totally not useable. Shipping this back to get my refund was also a nightmare. Thumbs down all around.	*refund nightmare. Thumbs .*

Table 2: A proper review before and after removing stopwords

With Stopwords	**Without Stopwords**
Eazy 2 use mine today I got it	*Eazy 2 use today got*

Table 3: An example of bad spelling and improper sentence structure

C. Synthesis and Visualization of Information

As stated above, using our manually classified reviews with the removal of stopwords to train and test the model, we received an average recall score of 0.3889, precision score of 0.3802, and F-score of 0.3755 compared to the manually classified reviews that kept stopwords which received 0.3443, 0.3389, and 0.3291 respectively. Figures 8, 9 and 10 are a visual representation for tests 1 and 2 of how the score changed over time as each review was tested. In the beginning, there is not a lot of information in our confusion matrix for the multi-class metric system so the result is either close to one or zero but as more reviews are tested, the calculation becomes more accurate than the final value.

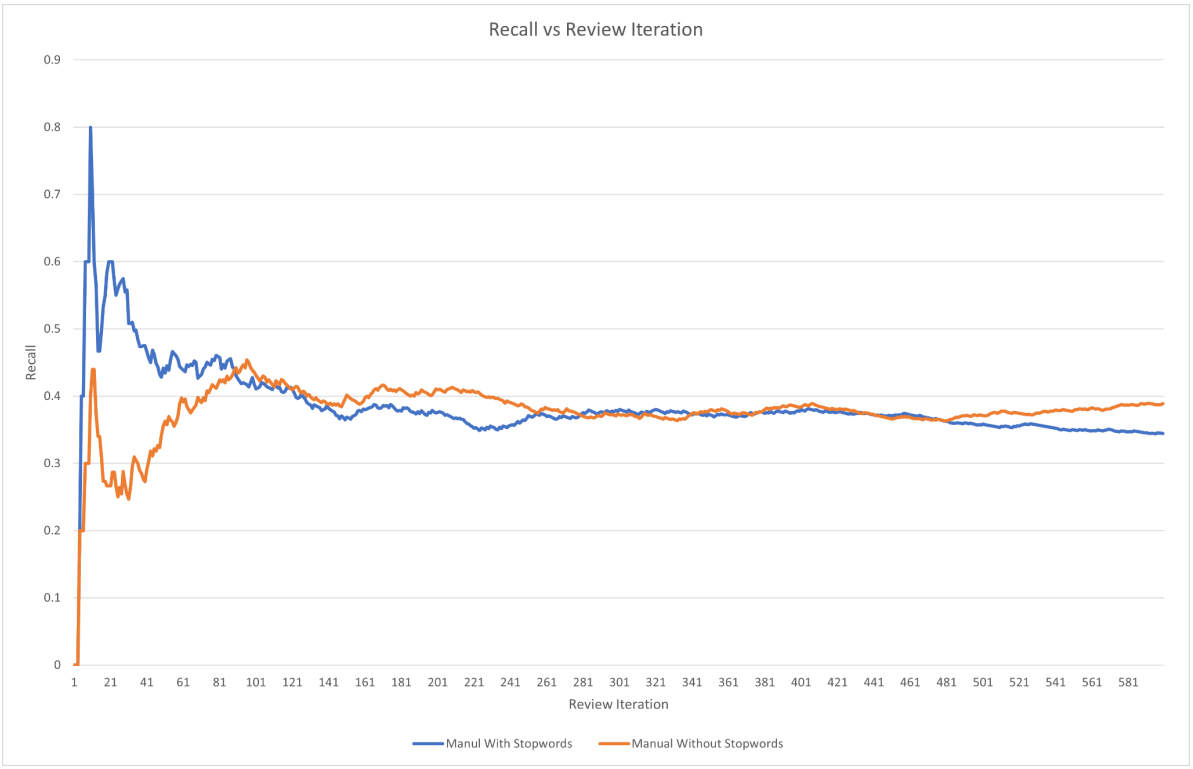

Figure 8: Graph of Recall vs Review Iteration for the Manually Rated Dataset

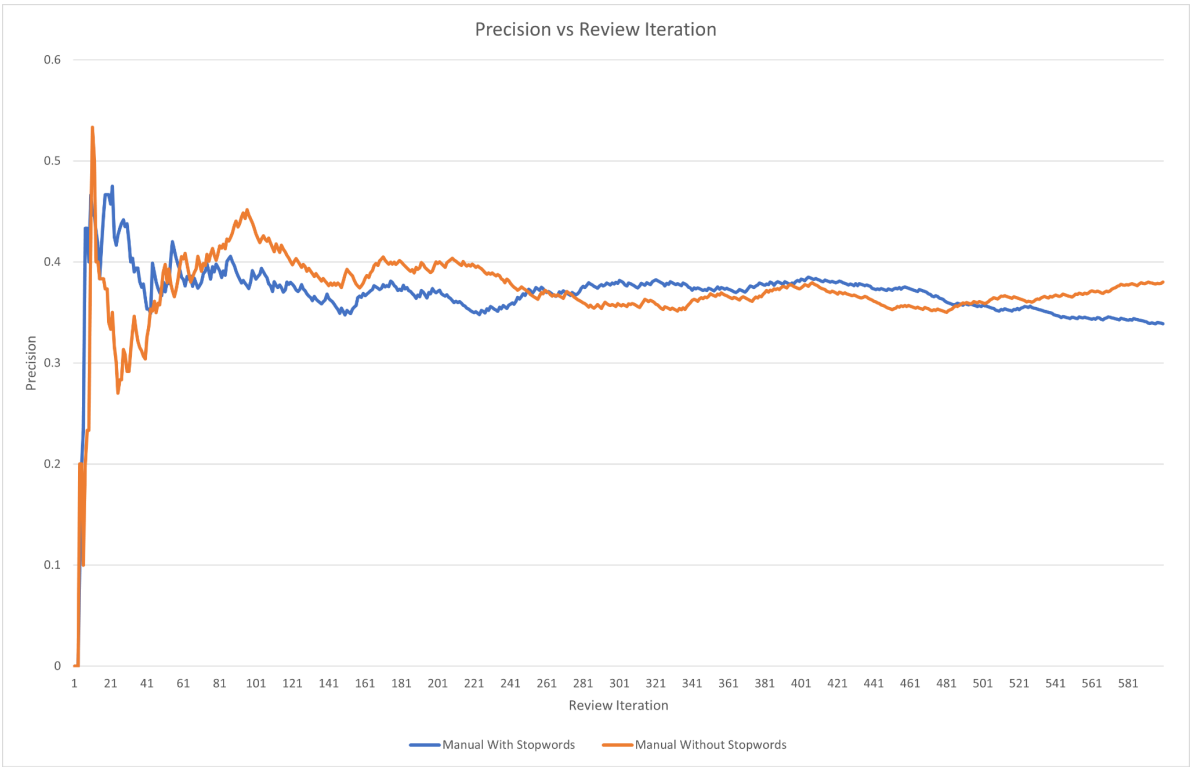

Figure 9: Graph of Precision vs Review Iteration for the Manually Rated Dataset

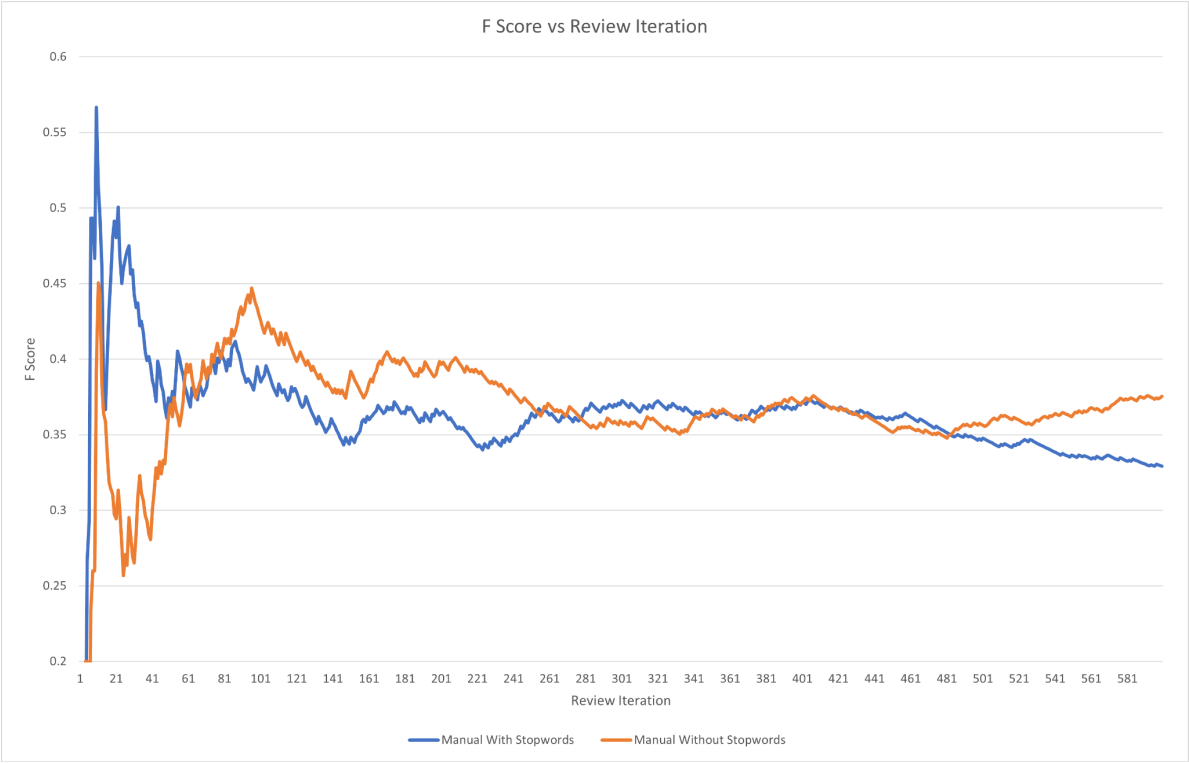

Figure 10: Graph of F-Score vs Review Iteration for the Manually Rated Dataset

The figures 11, 12 and 13 are a visual representation of the precision, recall, and F-score for tests 3, 4, 5, and 6. There were 25,000 reviews tested in these figures and from the graphs that value at around review #10,000 was very similar to the final value. Interestingly, when testing our model that was trained with our manually classified reviews we achieved better results versus the model that was trained off of the original review ratings. This seems reasonable since our rated reviews allow for a more accurate training process with a similar sentence structure. In figure 11, test 6, which was testing the model trained with the original rated reviews yielded the best precision, recall, and F-score values. Interestingly, test 3, where we trained the model on the manually classified reviews without stopwords, had the lowest precision, recall, and F-score. When testing the larger dataset of original rated reviews, we noticed that the test without

stopwords resulted in a lower F-score but there was not a pattern for the accuracy. Running the test again would result in the same information, but if we retrained the model and retested we could have been given different results and in this case, we could have averaged the results together.

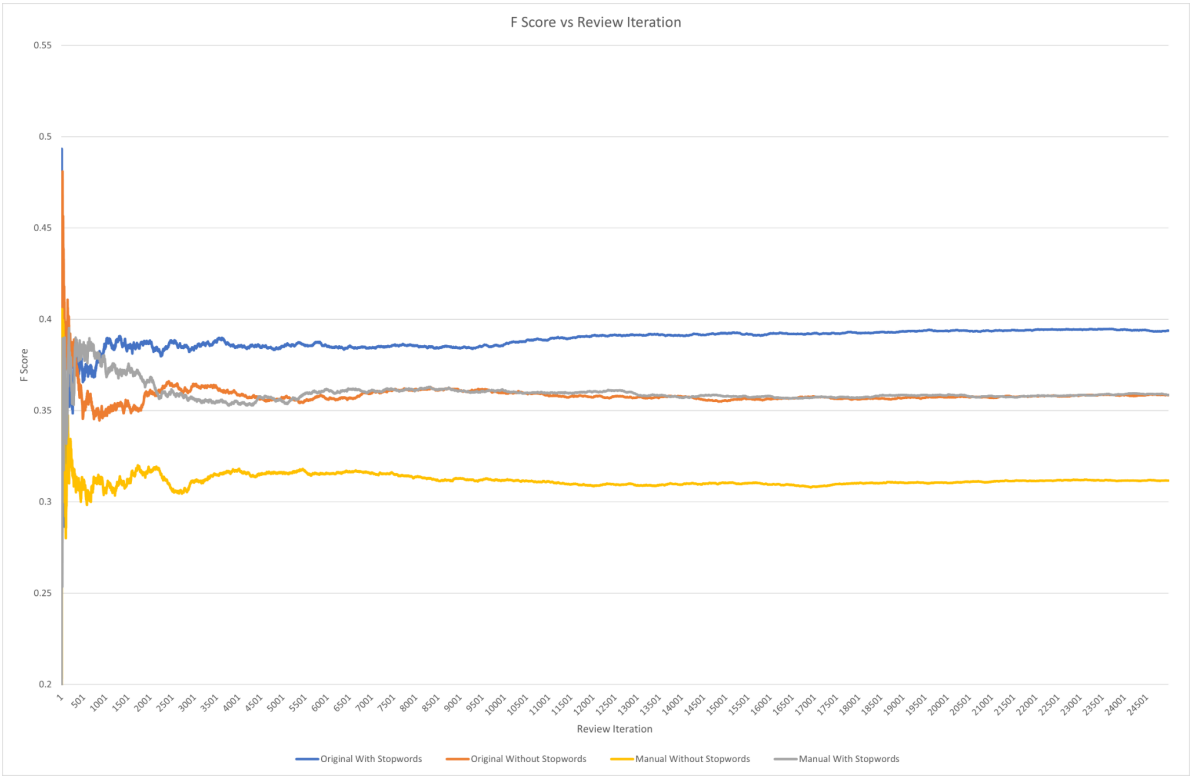

Figure 11: Graph of F-Score vs Review Iteration for the Original Rated Dataset

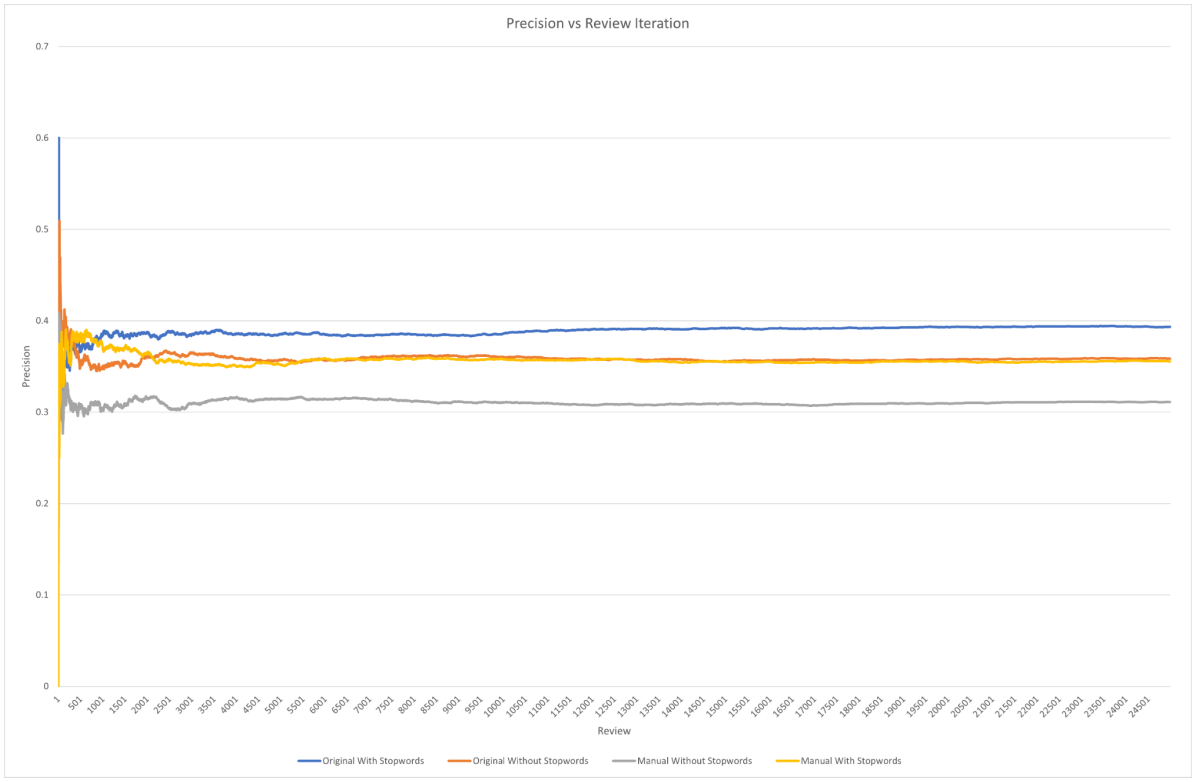

Figure 12: Graph of Precision vs Review Iteration for the Original Rated Dataset

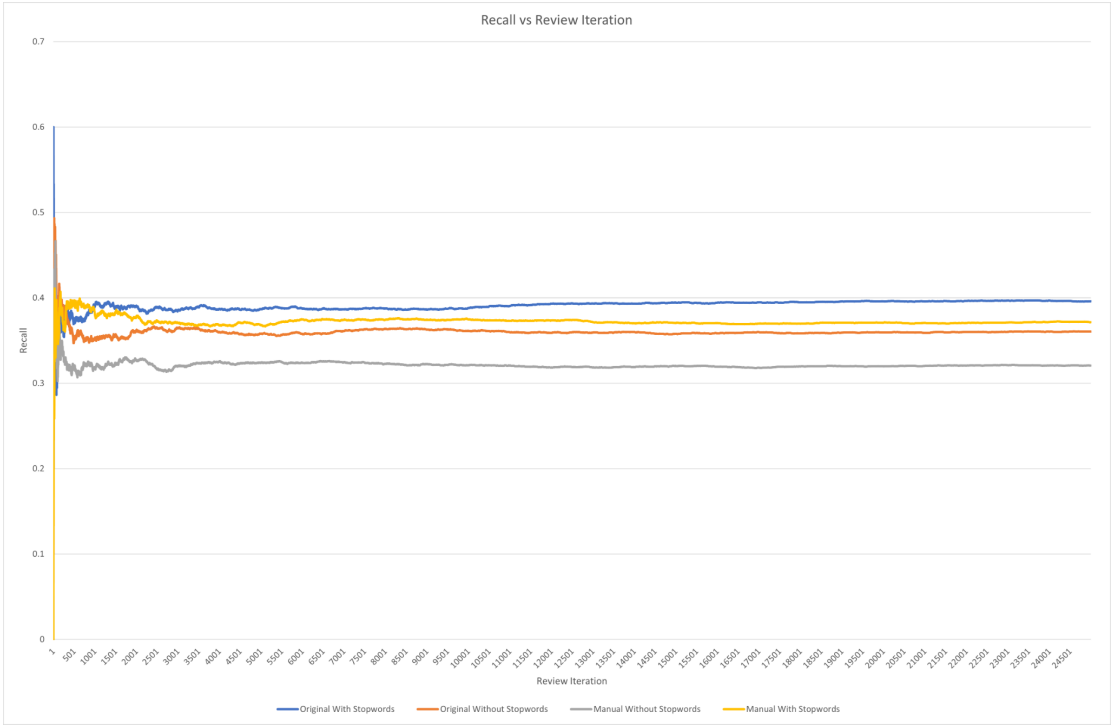

Figure 13: Graph of Recall vs Review Iteration for the Original Rated Dataset

VIII. Conclusion

In conclusion, based on research articles related to our field of study we were able to generate enough information to start developing a model to train and test reviews through. This model was able to successfully test over 25,000 reviews and provide an organized .csv file with all relevant information. As well, the program has the ability to take in a single review on a command line and quickly compute the sentiment in the given review. In testing, we concluded to have a real-world accuracy on average above 75% which we believe to be fair. We feel satisfied with the developed model, even though it is not as accurate as we had hoped when compared to the rating given to each review, the predicted ratings were still within reason. Especially when accounting for the difference of 1 star and even 2 stars of variance. If given more time, we feel that we could have improved the accuracy of the program either by implementing more features and algorithms that would account for the aforementioned variables or by training the model more with a wider variety of writing styles. We feel that with more adjustments, this model would be useful in the e-commerce field, as it allows companies to develop a deeper understanding of what the customer wants. If we had more time, we could have used our model on a cloud platform such as Amazon Web Services and which would allow us to create an API for further integration and ease of use since now it is not just a command-line tool but could be used in a web browser. In the end, the customer has the final say on where they shop, and if a company is able to improve their product and customer service based on customer feedback found when analyzing the various rated reviews, gives a competitive edge over other companies. It would have been interesting to have tested a larger dataset of reviews that related

to a number of different categories instead of just one, in this case, appliances. As a result, the project would have a larger use case without having to be refined and retrained further.

IX. Self-Reflective Report

A. Nicholas Imperius

I believe that our project can prove to be valuable to many businesses that are struggling with receiving and organizing customer feedback to further improve their products. A big challenge throughout the development process was how to implement the preprocessing and classifying techniques into python code. In addition, researching other projects to further gain an understanding of what type of testing was needed to be done was new to me. I managed to enhance my learning of NLP algorithms and techniques used throughout the industry. I feel as though our teamwork became stronger once we started developing our prototype since we were working on it altogether, bouncing ideas off each other trying to come up with the best possible solution.

B. Jimmy Tsang

In my opinion, I feel that our program was a success. Although our results determined that our model was not the most accurate, it was still a great opportunity to expand my knowledge of software development. Most notably, it was refreshing to learn about NLP and all the algorithms that can be used to process and classify large datasets of linguistic data. However, there was some difficulty encountered while navigating through these language processing algorithms, but this was cleared up through thorough research of scholarly articles and videos and explanations from my group members. Overall, I feel that the study and use of NLP can have

a major impact on the e-commerce industry and if given the chance, I would love to work on a similar project in the future.

C. Kristopher Poulin

In my personal opinion I believe that our project was very successful, throughout the experience I was able to expand my knowledge on NLP algorithms, as well as the basics of artificial intelligence programming with python. I believe that the idea behind the project offers great value to online businesses, as manual analysis of each review would take weeks in comparison to the reviewer model. A big challenge I faced during this project was categorizing all the new information I learned from the internet and scholarly sources, since there was so much information to absorb, it made it difficult to see the full picture and understand logically how each algorithm fits into our python program. As a team, our group worked very well together, this came naturally as we have worked together in previous courses.

X. References

[1] J. Jin, P. Ji, C.K. Kwong, “What makes consumers unsatisfied with your products: Review analysis at a fine-grained level,” Engineering Applications of Artificial Intelligence, Volume 47, 2016, pp. 38-48, ISSN 0952-1976, doi: 10.1016/j.engappai.2015.05.006.

[2] C. Chauhan and S. Sehgal, "Sentiment analysis on product reviews," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2017, pp. 26-31, doi: 10.1109/CCAA.2017.8229825.

[3] P. Chitra, T.S. Karthik, S. Nithya, J. Jacinth Poornima, J. Srinivas Rao, M. Upadhyaya, K. Jayaram Kumar, R. Geethamani, T.C. Manjunath, “Sentiment analysis of Product Feedback using natural language,” 2020, doi: 10.1016/j.matpr.2020.12.1061.

[4] Y. A. Solangi, Z. A. Solangi, S. Aarain, A. Abro, G. A. Mallah and A. Shah, "Review on Natural Language Processing (NLP) and Its Toolkits for Opinion Mining and Sentiment Analysis," 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS), Bangkok, Thailand, 2018, pp. 1-4, doi: 10.1109/ICETAS.2018.8629198.

[5] C.H. Lai, C.Y. Hsu, “Rating prediction based on combination of review mining and user preference analysis,” Information Systems, Vol. 99, 2021, 101742, ISSN 0306-4379, doi: 10.1016/j.is.2021.101742.

[6] C. C. Hnin, N. Naw and A. Win, "Aspect Level Opinion Mining for Hotel Reviews in Myanmar Language," 2018 IEEE International Conference on Agents (ICA), Singapore, 2018, pp. 132-135, doi: 10.1109/AGENTS.2018.8460040.

[7] K. Zvarevashe and O. O. Olugbara, "A framework for sentiment analysis with opinion mining of hotel reviews," 2018 Conference on Information Communications Technology and Society (ICTAS), Durban, South Africa, 2018, pp. 1-4, doi: 10.1109/ICTAS.2018.8368746.

[8] M. Bibi et al., "Class Association and Attribute Relevancy Based Imputation Algorithm to Reduce Twitter Data for Optimal Sentiment Analysis," in IEEE Access, vol. 7, pp. 136535-136544, 2019, doi: 10.1109/ACCESS.2019.2942112.

[9] F. M. Barcala, J. Vilares, M. A. Alonso, J. Grana and M. Vilares, "Tokenization and proper noun recognition for information retrieval," Proceedings. 13th International Workshop on Database and Expert Systems Applications, Aix-en-Provence, France, 2002, pp. 246-250, doi: 10.1109/DEXA.2002.1045906.

Appendix A

A. Meeting Minutes

Date: January 23rd

- The group assembled for a short meeting to discuss possible topics as well as brainstorm questions to ask our supervisor, Dr. Rachid Benlamri.

Date: January 26th

- A meeting is conducted with Dr. Benlamri about our project ideas. After discussing the outline of each of our ideas, we ultimately decided on developing a model that would analyze the sentiment behind customer reviews. The next steps were explored and Dr. Benlamri took his leave. Afterwards, the group developed the formal project proposal.

Date: February 5th

- Upcoming tasks and goals were laid out, as well as a due date. Work on the upcoming progress report was conducted.

Date: February 23rd

- A meeting was held to discuss the scholarly articles each individual group member collected over the study break. The progress report was then completed and await further feedback from Dr. Benlamri.

Date: March 3rd

- Development of the first oral presentation began. The group began summarizing information from the progress report into a presentable form for the presentation.

Date: March 6th

- With suggestions for improvements from Dr. Benlamri, the group convened to complete the presentation draft. Upon completion, the group then inquired about further changes with Dr. Benlamri

Date: March 9th

- Upon further discussion with Dr. Benlamri, the group resolved any faults that were discovered and after the last consultation, the oral presentation was submitted.

Date: March 18th

- A meeting was conducted with Dr. Benlamri about feedback on previously submitted work, including what we did well on and recommendations for improvement.
- Additionally, the next steps were and suggestions for how the prototype could be developed were discussed.

Date: March 25th

- Meeting to begin the installation of spaCy and begin creating the prototype
- Created scripts to parse through the datasets and create batches for us to manually classify

Date: March 27th

- Created a GitHub repository for easier grouping of each other code
- Worked on testing the training of a model with a small sample size to verify it worked properly before moving to the larger datasets

Date: April 2nd

- Meeting to see what progress we have made on manually classifying reviews

- Discuss possible improvements that could be made to our prototype

Date: April 8th

- A meeting was conducted with Dr. Benlamri about how to represent our results and what to include in the final presentation.

Date: April 10th

- Trained 4 different models based on the dataset we manually classified
- Reviewed the training data and test with single reviews to review accuracy

Date: April 12th

- Constructed the structure of the report and started to work on what we needed to put in it
- Created graphs based on the results from testing

Date: April 13th

- Finishing touches on the final report and final presentation
- Draft of the presentation was sent to Dr. Benlamri to verify if we were missing anything

Date: April 14th

- Incorporate suggestions made by Dr. Benlamri for the presentation
- Finished final report
- Rehearsed our final presentation
- Went through project code one last time to verify everything worked correctly