Consensus Analysis of Amazon Product Reviews

Smit Anand
North Carolina State University
Email: sanand2@ncsu.edu

Mahnaz Behroozi
North Carolina State University
Email: mbehroo@ncsu.edu

Erick Draayer
North Carolina State University
Email: ecdraaye@ncsu.edu

Shama Yazdani
North Carolina State University
Email: syazdan@ncsu.edu

***Abstract*—The online shopping industry has seen remarkable growth in the past decade. As more customers shift from in-store shopping experiences to online browsing, there is greater need for a way to distinguish between high quality products and junk. We explore the idea of using consensus analysis on the worlds number one ecommerce store, Amazon. We first explore three different approaches of building our own learner that can detect if a customers review is negative or positive as well as investigate existing systems. We then build our system around the most accurate learner and deploy it in three different ways: Web application, chrome extension, and android app. Lastly, we do a user evaluation and observe our participants using each iteration of our system and have them fill out a brief survey on what they thought of our system.**

***Index Terms*—Consensus Analysis, Amazon, e-shopping**

I. Introduction

As we progress through the Information Age, computers become more ingrained in everyday tasks and transform the way we complete them. Perhaps one of the biggest changes has been in how we shop for and buy goods. For example, whereas fifteen years ago it seemed that clothing items would never sell online, they now have a sizeable audience of online consumers. Large department stores such as Sears, Macys, and Dicks Sporting Goods are just a few among many that are failing to survive as more and more people look to online retailers for goods. Consumers are moving more towards online shopping, and thus moving towards a customer experience which is in many ways different from the traditional in-store experience. As part of this new experience, many consumers find themselves struggling to identify a quality product, especially if the product comes from a retailer with little to no reputation. In this situation, a person usually resorts to reading product reviews left by other people but this process can be long and tedious. We aim to provide a faster alternative to this problem by using consensus analysis to process a massive amount of user reviews at once, determine how positive and negative each review is, and then re-rate the product based on our overall average of the reviews.

In January, we conducted a pilot survey to characterize the shopping habits of online consumers, including: the most popular websites for shopping, important factors in consumers decisions about whether to purchase a product, which products consumers purchase most often online, and the importance of other users reviews to the consumer. We discuss the results of our survey in more detail in Section 2.1 of this report. Based on the results of the survey, we decided to build a consensus analysis system around Amazon. This system is meant to detect whether a product has majority positive, negative, or neutral reviews and assign a new rating to the product accordingly. We explore three different types of learners upon which to base our system. These include Naive Bayes, SVM, and Bernoulli Naive Bayes. Section 2.2 contains more information about the data used to construct our system. Sections 3.1 and 3.2 detail the construction and results of our three approaches to consensus analysis and problems encountered. We also explore a system which employs natural language processing by IBMs Bluemix AlchemyAPI, and compare this system to our learner-based system In Section 3.3, we describe IBM Bluemix AlchemyAPI and its performance in classifying the data.

We delivered our consensus analysis system to users on three different platforms: a web application, a Google Chrome extension, and an Android app. We discuss these three implementations in Section 4. Finally, we conducted a survey of our three implementations to assess user satisfaction level. We surveyed users to determine the ease of use for our system, which of the three implementations users liked best, the overall quality of experience with our system, and whether users trust the idea of consensus analysis in general. The methodology and results of our survey are detailed in Section 5.

II. Background

A. Initial Survey

We created an online survey using Google Forms. Our survey contained ten questions. Users were asked how often they use three online retailers: Best Buy, Amazon, and Yelp. These three stores were used because they are the only retailers to make their datasets of reviews for product and location available. We also asked participants how important product reviews are to them when deciding whether to buy a product. We then asked them how frequently they shop online and what types of products they generally buy online. In total we had twenty one participants for our online survey

Our survey showed Amazon to be the most popular online retailer, and also that customers value Amazon reviews more than reviews from other websites. Of those surveyed, 95.2% found customer reviews important or very important when deciding whether to buy a product. We wanted to build a consensus analysis system that takes these three survey findings into account.

TABLE I
COMPARISON OF DIFFERENT CLASSIFIERS

Classifier	Mean Accuracy (%)	Std (%)
Naive Bayes	74	4
SVM	50	8
Bernoulli Naive Bayes	73	3

B. Data

To build our system we used data publicly provided by Amazon. The data consists of 1,000 Amazon product reviews with corresponding labels describing whether the review is positive or negative (1 = positive, 0 = negative). The reviews are provided through a standard .txt file that has one review per line, with the number label at the end of each review. The data [1][2] can be accessed here: http://jmcauley.ucsd.edu/data/amazon/

C. Proposed Solution

We decided to construct a learner to detect the positiveness or negativeness of many user reviews simultaneously. We chose three different approaches to build and test our learner: Naive Bayes, SVM, and Bernoulli Naive Bayes. We also wanted to explore existing sentiment analysis systems, so we decided to test our learners against IBMs Bluemix AlchemyAPI. Once we had determined the best analysis system, we would then launch our system across three different platforms: a Google Chrome extension, a web app, and an Android app.

III. METHODS

A. Our approach to sentiment analysis

We implemented our own method of natural language processing using Python NLTK. We wrote our code which consists of parsing, word extraction, and feature extraction. We applied this learner to a set of 1000 labeled reviews consisting of 500 negative reviews (labeled as zero) and 500 positive reviews (labeled as one). For evaluating the performance of our algorithm, we applied three different classifiers including Naive Bayes classifier, SVM and Bernoulli Naive Bayes classifier with the help of 10-fold cross validation (CV) technique. We obtained the following results, where mean accuracy is the average of ten accuracies obtained from applying 10-fold CV and std is the standard deviation of these 10 accuracies, shown in Table1.

From the results we can infer that Naive Bayes classifier is the most accurate and robust classifier in comparison to SVM and Bernoulli Naive Bayes classifier. SVM did not perform well on our dataset and seems to be no better than a learner that randomly guesses. We concluded that it must be because our data points in the problem space were not linearly separable. If an SVM approach were to be taken in the future, it should be though a kernel SVM instead. Fig. 1 shows a screen shot of our implemented web application.

We implemented our Naive Bayes model in Flask through a web page. Flask is a webform framework which can be used with Python to implement a web site. We gave an option

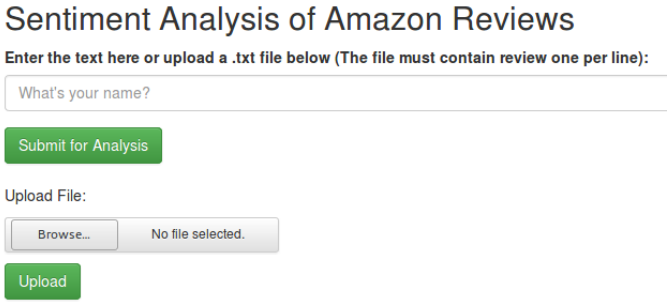

Fig. 1. Our Learner Running with Flask

to analyze a single review or analyze multiple reviews. Our webpage consisted of a textfield in which users can enter their review(s) to be analyzed. When the user presses the analyze button, the result is reported as a 1 (positive) or 0 (negative). Each time the webpage is initialized, the model must be retrained. Because of this, our system is very slow when using Naive Bayes and so we opted not to present this approach when conducting our user evaluations. Nonetheless, we designed our system so that in the future, we could still use our Naive Bayes learner if we had a server to host it on. We also designed this system to allow us to easily test the accuracy of any type of learner we were using. We can classify a large amount of test reviews and store the results in a text file for testing accuracy immediately thereafter.

B. Roadblock

Though our approach gave good results the initial roadblock was deploying our server. Even after deployment the query to server took lot of time as every query initiated re-training of our model, also we always needed a system on which the server will be running and that created dependency for all other systems on this. To analyze a single review it took around 6 seconds and to analyze the file it took around 10 secs which made our application work slow. We needed something which could give result faster and better and reduce the dependency on our system.

Another major problem we faced came when we found out Amazon shut down all the APIs of python as well as others created to retrieve user reviews from their website. Amazon does offer APIs themselves through javascript on their website but it costed more money than what we were willing to spend and we would need to find a way to adapt our learner to be used in javascript. Even if we did decide to go this route, Amazon is still very restrictive about grabbing massive amounts of user reviews at one time so there was no guarantee that this would be a viable solution.

With these two major problems facing us we decided to look for already present consensus analysis systems that we could use for our application. After a quick search we found that IBM has made Watsons natural language capabilities, including sentiment analysis, public through Alchemy API.

C. IBM Bluemix AlchemyAPI

AlchemyLanguage is a collection of natural language processing APIs that helps understand sentiment, keywords, entities, high-level concepts and more[3]. It is provided by the IBM Bluemix which is an is an implementation of IBM's Open Cloud Architecture based on Cloud Foundry, an open source Platform as a Service (PaaS). Alchemy API can take input as text, document or a url and can provide sophisticated natural language processing. For input such as HTML or url it automatically removes the ads and other unwanted texts leaving only the important content to be analyzed. To access Alchemy API an authentication is required which is done by passing the API key as parameter when calling the service. This API key can be obtained by signing up for IBM Bluemix, it also gives 30 days of free access to Alchemy API with a restriction of 1000 queries per day.

The sentiment analysis works for text, HTML as well as web page and supports Arabic, English, French, German, Italian, Portuguese, Russian and Spanish. After analysis the sentiment of the input it returns AlchemyAPIs dynamic approach enables the system to have a much deeper understanding of text. AlchemyAPI works great for following[4]: 1. Negations which are reverse the polarity of sentiment. Example: this product is great and this product is not great. 2. Amplifiers which are words which increases the intensity of the sentence. Example very. 3. Diminishers which are words which decrease the intensity of the sentence. Example somewhat. The above parameter determines the intensity score of detected sentiment. Example the intensity of This coffee is a little sweet, and, This coffee is totally disgusting, vary greatly.

In a paper[5] by Meehan et al (2013), Context-Aware Intelligent Recommendation System for Tourism1 , showed that AlchemyAPI achieved 86.01 percent accuracy level after manual testing performed on a corpus of 5370 tweets. We tested the AlchemyAPI on the our labelled dataset of 1000 amazon reviews and got the accuracy of 85.8%. Bluemix AlchemyAPI allowed us to get around having to retrain our model every time we wanted to analysis reviews as well as provide us with a very high accuracy so we decided to build our system around it.

IV. Method Implementation

Prior research has shown that Amazon has a customer base of 304 million people, with an estimated 70% of users preferring to access Amazon via their smartphone and the rest favoring a home computer. From the above statistics it is very clear that the majority of customers use their smartphone for shopping from amazon, but the remaining 30% cannot be ignored since it is still a large group of people. So we decided to implement solutions for both types of customers. Mary Meeker's 2016 Internet Trends report [6] suggest that approximately 80% of smartphone users use Android OS and the remaining use iPhone or other operating systems. Thus it was clear to us that we should develop an Android application. We also implement our analysis of reviews through Chrome extension and web application, totalling to three different platforms.

A. Web Application

This approach is designed using Python. The UI consists of a textfield which takes a single review from user to analyze it. When clicking the "submit for analysis" button the request is sent to IBM Bluemix (with the use of AlchemyAPI) server and the received response is displayed as the sentiment of the review and the overall score of the sentiment is displayed. Alternatively, the user also has the option to enter the product URL that links directly to a product and our system will perform consensus analysis on all text from that page and report the overall sentiment of the product, give the score of sentiment as well as re-rate the star rating as per the analysis.

The interface of the web page is shown in Fig2.

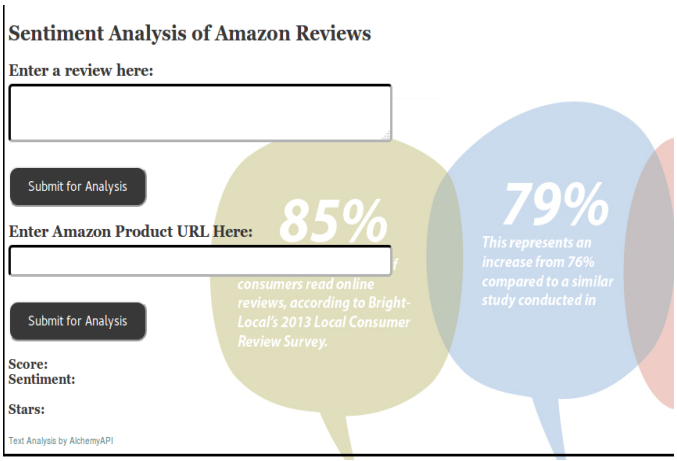

Fig. 2. Appearance of Working Web Application

The above fig. 2 gives a working idea of our web application.

B. Chrome Extension

The chrome extension is very similar to the web app. Users have the option of single review or multiple (bulk) review analysis. And like the web application the user can analyze a single review or pass the url of the product to get the overall sentiment of reviews of that product. The advantage of this approach is that the user does not need to navigate to a new window or tab to access the review software. Instead, users click the extension icon on their toolbar and the extension appears as a small window. Again, users can copy/paste a review or enter a URL into the extension window to see the overall sentiment of the product. This aims to provide the user ease of access to our application. And as aimed this solution was most liked by users in user evaluation which is explained late in the report.

In Figure 3 you can see the User Interface of the Chrome Extension and shows how easily you can copy and paste the web address. The app will analyze the overall sentiment of the reviews of the Amazon product and re-rate the product without moving to another tab or window.

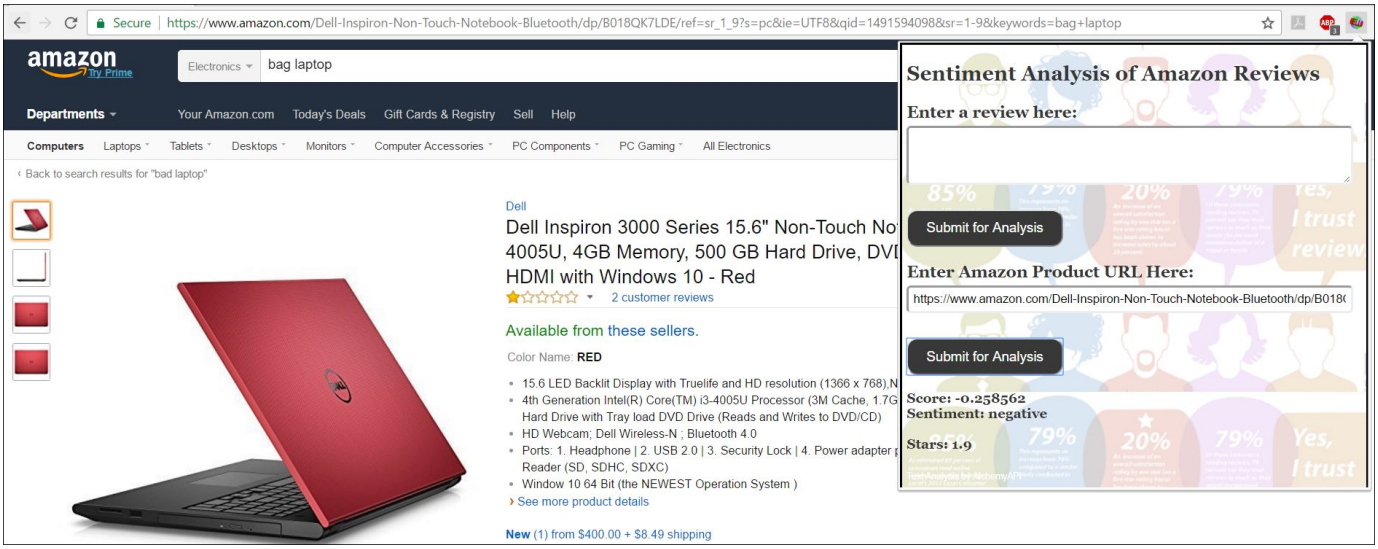

Fig. 3. Chrome Extension as it Appears on a User's Screen

C. Android Application

The android application was designed using Android Studio and uses Java as the programming language. Like the other approaches, it uses the IBM BLuemix AlchemyAPI for sentiment analysis of the reviews. The application is available to download at the github link of this project as ReviewAnalysis-App.apk and requires android version 5.0 (lollipop) or above to function.

The welcome screen of the application is shown in figure 4 which gives two options to the user: analyze a single review, analyze multiple of reviews or to analyze a product review through the product URL. Upon clicking the button to analyze single review the screen shown in fig 4 appears which asks the user for the review and upon clicking the analyze button the application sends request to Alchemy API server and upon receiving the sentiment from the server displays it to the user as seen in figure 4.

Upon choosing the option to analyze multiple reviews the screen shown in figure 5 comes up. To analyze multiple reviews the text file containing reviews must be kept the the location /sdcard and user must provide the app access to read and write the storage in device. Upon clicking the button to upload reviews from file the file is selected content of the file gets displayed and then the user can send the file for analysis. Upon receipt of response from the server the overall sentiment of the bulk review is displayed to the user as shown in figure 5.

The android application also gives the option to analyze the review of product through amazon url. the user has to enter the url of amazon product which can be obtained by share option given while viewing the product using amazon app and and copying it to the clipboard. The clipboard content then can be pasted in the url section and the product can be analyzed by clicking the button as shown in figure 6.

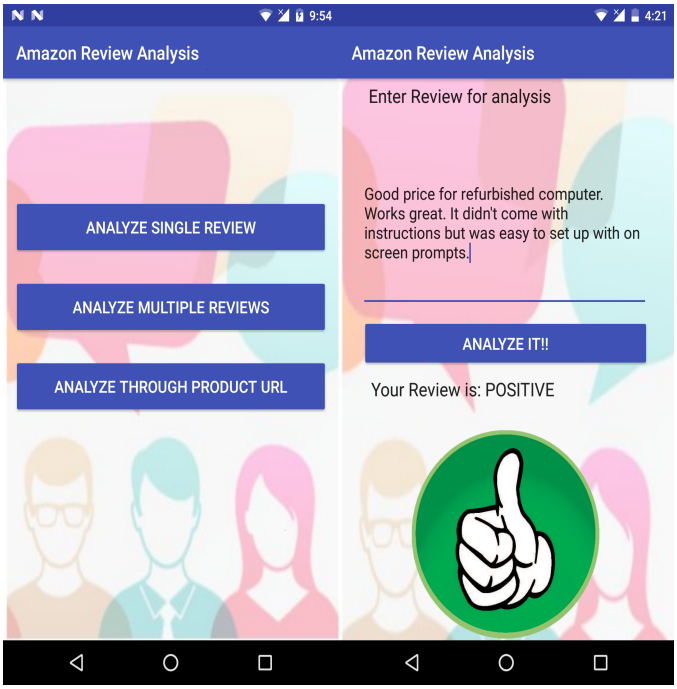

Fig. 4. Selection Screen and Single Review.

V. Results

A. User Evaluation

For the sake of understanding how successful our solutions would be from their users point of view, we conducted a user survey which targeted the ease of use, user friendliness and accuracy. We asked the users to input whatever review sentences or paragraphs in either web app, Chrome extension or android app without any restrictions and see if our software applications could successfully classify their input as positive or negative. We also provided a link of bulk review URL to

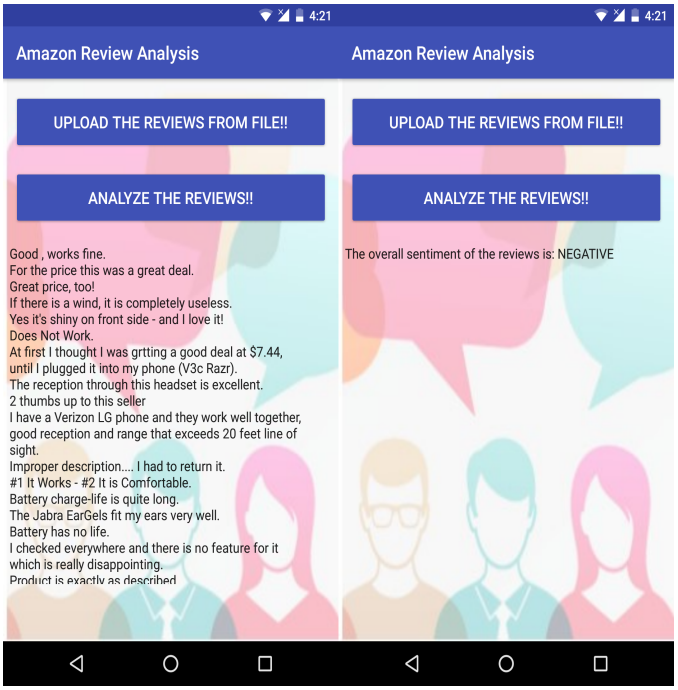

Fig. 5. Multi-Review Selection.

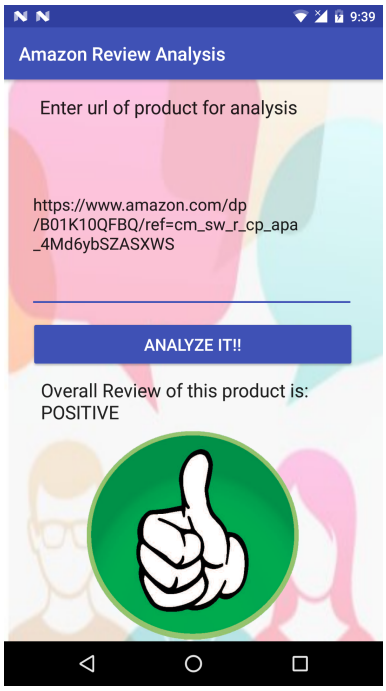

Fig. 6. Analyze through prduct URL

the users to show them that our system is also capable of analysing the reviews fetched from a particular web page, although right now Amazon does not give us access to fetch their products pages. We invited 17 users to our user evaluation session and provided our software applications to them and asked them the following questions for each application after they tested them:

Q1: Did the result match the tone of your review? (Yes, No), Q2: How easy was it to learn and use our application? (from 1 to 5), Q3: Rate your experience with our software. (from 1 to 5), Q4: Describe the look of our software application. (from 1 to 5),

And three wrap-up questions:

Q1: Which app was the easiest one to use?, Q2: Which one did you like the best?Why?, Q3: In general, would you use sentiment analysis to help you purchase something?
Results from questions 1-4:

TABLE II
RESULTS FROM QUESTIONS 1-4:

	Web App	Chrome	Android App
% Correct analyses	0.882	1.00	1.00
Avg ease of use rating (out of 5)	4.59	4.88	4.29
Avg experience rating (out of 5)	4.53	4.53	4.12
Avg appearance rating (out of 5)	4.12	4.29	4.12

B. Best Approach

From the results we can see that users picked the Chrome extension as the easiest to use and overall liked it better than the other two. Our chrome extension also had the best apperence rating. Users seemed to be more satisfied with the experience of using sentiment analysis through the Chrome extension and Web application. Clearly the Chrome extension was the best platform in terms of looks and feel compared to our other two platforms. All in all, 88.2% of our users were interested in using sentiment analysis to help them make online purchasing decisions(Fig. 7).

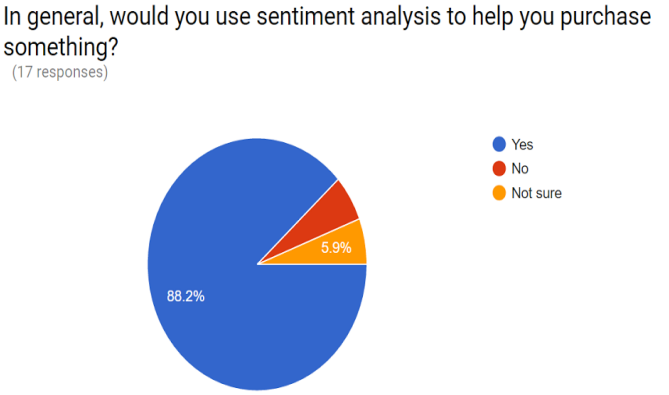

Fig. 7. Willingness to use Sentiment Analysis in Purchasing Decisions

VI. CONCLUSION

We created a consensus analysis system to assess Amazon product reviews. The system was designed to detect how

negative or positive a products reviews are on average, and to assign a new star rating to the product accordingly.

We tried three different approaches to create our system: Naive Bayes, SVM, and Bernoulli Naive Bayes. Using data provided by Amazon, we created a training set and a test set to fit and test the three systems. We found Naive Bayes to be the best, with accuracy 74.0%. We also tested IBMs Watson natural language processing through Bluemix AlchemyAPI on our data. Bluemix allowed us to work around the restrictions Amazon has placed on APIs and has higher accuracy than our system, at 85.8% accuracy. For this reason, we decided to use Bluemix for the consensus analysis in our final product.

We implemented consensus analysis using Bluemix across three different platforms: a Google Chrome extension, a web app, and an Android app. We conducted user evaluations on the three platforms to determine which implementation had the greatest ease of use and the best overall user experience. The Chrome extension proved to be the most popular, with the majority of users rating it as the easiest to use and the best overall experience when compared to the web app and Android app. We also showed that there is a general interest in a system like this to help purchase items online and its something worth developing further.

VII. Future Work

From our user evaluations, we discovered that there is still a lot of room left for improvement besides the full implementation with Amazon. People overall had different complaints about each implementation style and feel. But a consistent complaint we had was that we needed to make the results more apparent throughout all three of our implementations.

As discussed above, the Chrome extension was overall the easiest for people to use but they felt that the look of it was a bit too busy and that we should go for a simpler design. They also wanted the icon for the chrome extension to be more related to our system, as of right now, we are using generic one that can be found on any chrome tutorial website. They also felt that we should display the stars as actual stars instead of a number. These are all small and easy changes to make that would help help with the user experience.

For our web application people wanted it to be centered on the screen, instead of everything aligned to the left. Also, much like the chrome extension, they felt we should display the stars as actual stars instead of a number. People also complained that our web application had a bit of a plain look to it and it made it boring to use. Again, these are easy and small changes to fix, but are still important if we ever wanted to have a large user base for our system.

Many people felt our android app implementation was the hardest to learn and the least intuitive. People wanted the star system we had in the other two implementations. They also suggested that we try integrating this with the Amazon app, although we are not sure that something like that can even be done given that we are working on a mobile OS. Lastly we had complaints that the buttons did not look like buttons and they thought they were headers. We should look into changing the UI so that it is more intuitive and easier to use.

In future we have thought to further implement it using Django (another python web interface framework) because flask is a very lightweight framework which does not give too much option of customization. Like you can only use one POST method to call for only one thing , either textbox or bulk upload, not both in the same program. So, you cannot implement both of them in the same program. For this we need to change our implementation to Django, so that we can implement several functionality in the same program.

Besides the look and feel of our android app, chrome extension, and web application, improvement with our learner could also be made to try and outperform Watsons natural language sentiment analysis. The main problem we faced with our learner is that we ran out of data to learn from, but we still managed to achieve a respectable accuracy rate of predicting sentiment of Amazon reviews. Perhaps a different approach we didnt think about or simply finding more data would enable us to surpass Watsons accuracy.

We would also like to explore the idea of expanding beyond Amazon and into other realms. We realized our system might be better suited for an environment like Twitter, where people cannot leave stars but still write semi-review like texts for things they have purchased or experienced. We think it would be relatively easy to modify our system to instead, or perhaps also, include reviews from Twitter. We could start by simply looking for hashtags that include the name of the product someone is interested in and gathering all the tweets and run them through our sentiment consensus. A system like this could take advantage of finding the sentiment on a product that just came out, since Amazon reviews do not usually appear in mass until one or two days after the product has already been out. Using Watson also allows the possibility of adding multi-language support so that we can analyze the sentiment of any review written in anything from Afrikaans to Xhosa.

Overall there are still many ways to improve the system in both the look and feel of each implementation as well as expanding the system as a whole. We feel our system has a lot of potential for helping users gauge the value of a product in a very quick and effective manner based on the reviews people are writing for that product. Based on the analysis from our user evaluation, we would probably focus on the chrome extension implementation exclusively, and improving the learner we developed to try and beat Watson. Lastly we would try to implement some sort of Twitter integration so that we could even get sentiment consensus for products that have have just been released.

References

[1] McAuley, Julian, et al. "Image-based recommendations on styles and substitutes." Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2015.

[2] McAuley, Julian, Rahul Pandey, and Jure Leskovec. "Inferring networks of substitutable and complementary products." Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015.

[3] https://www.ibm.com/watson/developercloud/alchemy-language.html

[4] https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=LBW03035USEN
[5] Meehan, Kevin, et al. ”Context-aware intelligent recommendation system for tourism.” Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on. IEEE, 2013
[6] http://www.kpcb.com/internet-trends